

UNIVERSITÀ DEGLI STUDI DI PADOVA  
FACOLTÀ DI INGEGNERIA



*Finito di scrivere il giorno 23 giugno 2006 utilizzando L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>*

UNIVERSITÀ DEGLI STUDI DI PADOVA  
FACOLTÀ DI INGEGNERIA

—  
DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

—  
CORSO DI PROGETTAZIONE DI SISTEMI DI CONTROLLO

PIANIFICAZIONE DI PERCORSO E  
CONTROLLO DI UN UNICICLO  
(PROGETTO NUMERO 10)

PROFESSORE: LUCA SCHENATO

STUDENTI: ENRICO BANDINU, MATTIA BRUSCHETTA,  
CARLO MACCHION (GRUPPO NUMERO 9)

ANNO ACCADEMICO 2005-2006



# Indice

<b>Introduzione</b>	<b>VII</b>
<b>1 Modellizzazione unicycle</b>	<b>1</b>
1.1 Modellizzazione cinematica . . . . .	2
1.2 Modellizzazione dinamica . . . . .	3
1.3 Dimostrazione anomalia del sistema[5] . . . . .	4
<b>2 Descrizione del khepera</b>	<b>7</b>
2.1 Caratteristiche fisiche . . . . .	7
2.2 Encoder . . . . .	7
2.3 Sistema di comunicazione e tempi . . . . .	8
2.4 Limiti velocità . . . . .	8
<b>3 Path Planing</b>	<b>9</b>
3.1 Generazione nel tempo . . . . .	9
3.1.1 Sinusoidale (punto-punto) . . . . .	12
3.1.2 Polinomiale (punto-punto) . . . . .	13
3.1.3 Limiti alla generazione di traiettorie nel tempo . . . . .	15
3.2 Generazione nello spazio . . . . .	17
3.2.1 Imposizione di velocità su traiettoria analitica . . . . .	17
3.2.2 Spline . . . . .	17
<b>4 Azioni di controllo</b>	<b>21</b>
4.1 Generazione azione feed-forward . . . . .	21
4.2 Controllori implementati . . . . .	21
4.2.1 Linear desing . . . . .	22
4.2.2 Nonlinear desing . . . . .	22
4.2.3 Dynamic feedback linearization . . . . .	22
<b>5 Simulazioni</b>	<b>25</b>
5.1 Modelli del sistema . . . . .	25
5.1.1 Modello continuo . . . . .	25
5.1.2 Modello discreto . . . . .	26
5.2 Azione Feed-Forward . . . . .	26
5.3 Azioni di controllo . . . . .	28

5.3.1	Linear design . . . . .	28
5.3.2	Non linear design . . . . .	29
5.3.3	Dinamic feedback linearization . . . . .	29
5.4	Limiti del controllo . . . . .	33
5.5	Cuspidi . . . . .	33
5.6	Verifica correttezza calcolo spline . . . . .	35
<b>6</b>	<b>Risultati sperimentali</b>	<b>39</b>
6.1	Problematiche sperimentali . . . . .	39
6.2	Feedforward . . . . .	40
6.3	Azioni di controllo . . . . .	40
6.3.1	Linear design . . . . .	40
6.3.2	Nonlinear desing . . . . .	42
6.3.3	Dinamic feedback linearization . . . . .	42
6.4	Limiti del controllo . . . . .	42
6.5	Cuspidi . . . . .	44
<b>7</b>	<b>Evoluzioni future</b>	<b>45</b>
	<b>Conclusioni</b>	<b>47</b>
	<b>Bibliografia</b>	<b>49</b>

# Introduzione

Il controllo di un autoveicolo è uno dei progetti più ambiziosi e soggetti ad un certo grado di difficoltà. Nell'esperienza presentata si è analizzato un modello semplificato di un veicolo completo, cioè si è analizzato un unicycle. Per unicycle si intende un mezzo a due ruote non sterzanti appartenente alla classe di sistemi non lineari detti anolonomi, cioè con vincoli istantanei sulle velocità, ma che possono comunque raggiungere qualsiasi configurazione nel piano. L'automa da noi utilizzato è denominato *Khepera*.

Il lavoro compiuto è descritto nei seguenti capitoli nel seguente ordine:

1. modellizzazione dell'unicycle e valutazione dei vincoli anolonomi
2. descrizione del sistema utilizzato e limitazioni
3. pianificazione di traiettorie utilizzando diverse tecniche
4. generazione dell'azione di controllo per percorrere le traiettorie generate
5. analisi dei risultati sperimentali ottenuti

Nella parte conclusiva verrà affrontata una breve trattazione del possibile lavoro che si potrebbe svolgere come continuazione dell'esperienza trattata ed una valutazione sulle difficoltà e le problematiche che si potrebbero incontrare in questa evoluzione.





# Capitolo 1

## Modellizzazione unicycle

Il modello del sistema da noi utilizzato consiste in 2 ruote indipendenti montate su di un asse comune (come illustrato in figura 1.1), ognuna delle quali può essere guidata indipendentemente dall'altra.

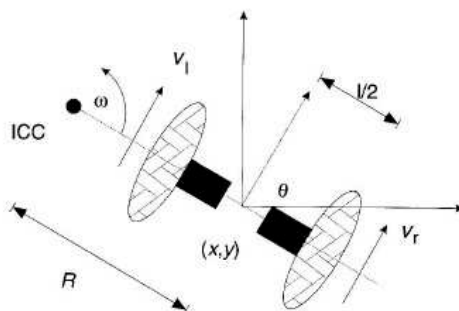


Figura 1.1: Modello del sistema

Variando le velocità delle ruote si può variare la traiettoria percorsa dal robot intorno al suo centro istantaneo di curvatura.

Possiamo scrivere le seguenti equazioni:

$$\omega (R + l/2) = V_r \quad (1.1)$$

$$\omega (R - l/2) = V_l \quad (1.2)$$

con  $l$  lunghezza dell'asse delle due ruote,  $V_r$  e  $V_l$  le velocità della ruota destra e sinistra e  $R$  la distanza dal punto di curvatura, calcolabile come:

$$R = \frac{l}{2} \frac{V_l + V_r}{V_l - V_r} \quad (1.3)$$

$$\omega = \frac{V_l - V_r}{l} \quad (1.4)$$

Si nota, da queste equazioni, il robot non può muoversi lungo la direzione dell'asse.

## 1.1 Modellizzazione cinematica

In figura 1.1, assumendo che il robot sia in una certa posizione  $(x, y)$ , con riferimento al sistema di riferimento 1.2, si ha un angolo  $\vartheta$  con l'asse di riferimento  $X$  del fixed frame. Assumeremo come posizione del robot il centro dell'asse che congiunge le due ruote. Agendo sulle velocità  $V_r$  e  $V_l$  possiamo porre l'automa in diverse posizioni e orientazioni dello spazio.

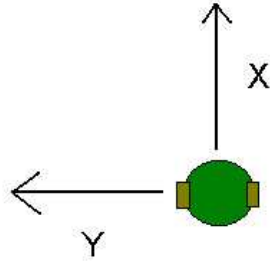


Figura 1.2: Sistema di riferimento adottato

Note le velocità ed utilizzando le equazioni 1.3 e 1.4 possiamo determinare il centro di rotazione del sistema:

$$ICC = [x - R\sin(\vartheta), y - R\cos(\vartheta)] \quad (1.5)$$

e al tempo  $t + \delta t$  il robot si posiziona in[?]:

$$\begin{bmatrix} x' \\ y' \\ \vartheta' \end{bmatrix} = \begin{bmatrix} \cos(\omega\delta t) & -\sin(\omega\delta t) & 0 \\ \sin(\omega\delta t) & \cos(\omega\delta t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - ICC_x \\ y - ICC_y \\ \vartheta \end{bmatrix} + \begin{bmatrix} ICC_x \\ ICC_y \\ \omega\delta t \end{bmatrix} \quad (1.6)$$

il movimento compiuto è presentato in figura 1.3.

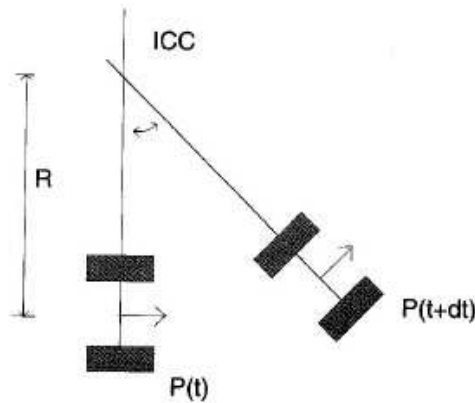


Figura 1.3: Movimento del robot

Utilizzando come coordinate generalizzate  $q = [x, y, \vartheta] \in Q = \mathfrak{R}^2 \times SO^1$  per  $n = 3$ . Il vincolo che le ruote non possano slittare nella direzione laterale è:

$$A(q)\dot{q} = \dot{x}\sin\vartheta - \dot{y}\cos\vartheta = 0 \quad (1.7)$$

Otteniamo perciò il modello cinematico del primo ordine[1]:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\vartheta} \end{bmatrix} = g_1(q)v + g_2(q)\omega = \begin{bmatrix} \cos\vartheta \\ \sin\vartheta \\ 0 \end{bmatrix} v + \begin{bmatrix} 0\vartheta \\ 0\vartheta \\ 1 \end{bmatrix} \omega \quad (1.8)$$

dove  $v$  e  $\omega$  (rispettivamente velocità lineare e velocità angolare del veicolo) sono presi come 2 ingressi di controllo.

Il rispettivo modello a catena risulta:

$$\dot{z}_1 = v_1 \quad (1.9)$$

$$\dot{z}_2 = v_2 \quad (1.10)$$

$$\dot{z}_3 = z_2 v_1 \quad (1.11)$$

dove:

$$z_1 = x \quad (1.12)$$

$$z_2 = \tan(\vartheta) \quad (1.13)$$

$$z_3 = y \quad (1.14)$$

con ingressi di comando:

$$u_1 = v_1 / \cos\vartheta \quad (1.15)$$

$$u_2 = v_2 \cos^2\vartheta \quad (1.16)$$

## 1.2 Modellizzazione dinamica

La modellizzazione dinamica del modello[1] è costruibile a partire da  $\eta = (x, y)$  e derivando nel tempo si ottiene che:

$$\dot{\eta} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \cos\vartheta \\ \sin\vartheta \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (1.17)$$

in cui si nota che solo  $v$  influisce su  $\dot{\eta}$ , mentre la velocità angolare  $\omega$  non fornisce informazioni nel modello del primo ordine. Per proseguire è necessario inserire un integratore (chiamato  $\xi$ ):

$$v = \xi, \quad \xi = a \Rightarrow \dot{\eta} = \xi \begin{bmatrix} \cos\vartheta \\ \sin\vartheta \end{bmatrix} \quad (1.18)$$

dove  $a$  è l'accelerazione dell'uniciclo. Derivando nuovamente si ottiene che:

$$\ddot{\eta} = \dot{\xi} \begin{bmatrix} \cos\vartheta \\ \sin\vartheta \end{bmatrix} + \xi \dot{\vartheta} \begin{bmatrix} -\sin\vartheta \\ \cos\vartheta \end{bmatrix} = \begin{bmatrix} \cos\vartheta & -\xi \sin\vartheta \\ \sin\vartheta & \xi \cos\vartheta \end{bmatrix} \begin{bmatrix} a \\ \omega \end{bmatrix} \quad (1.19)$$

con la matrice che risulta non singolare per  $\xi \neq 0$ , in tale modo, sotto questa assunzione si definisce:

$$\begin{bmatrix} a \\ \omega \end{bmatrix} = \begin{bmatrix} \cos\vartheta & -\xi \sin\vartheta \\ \sin\vartheta & \xi \cos\vartheta \end{bmatrix}^{-1} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (1.20)$$

ottenendo:

$$\ddot{\eta} = \begin{bmatrix} \ddot{\eta}_1 \\ \ddot{\eta}_2 \end{bmatrix} = u \quad (1.21)$$

Il risultato del compensatore dinamico:

$$\dot{\xi} = u_1 \cos\vartheta + u_2 \sin\vartheta \quad (1.22)$$

$$v = \xi \quad (1.23)$$

$$\omega = \frac{u_2 \cos\vartheta - u_1 \sin\vartheta}{\xi} \quad (1.24)$$

Essendo  $\xi \in \mathfrak{R}$ , nelle nuove coordinate abbiamo:

$$z_1 = x \quad (1.25)$$

$$z_2 = y \quad (1.26)$$

$$z_3 = \dot{x} = \xi \cos\vartheta \quad (1.27)$$

$$z_4 = \dot{y} = \xi \sin\vartheta \quad (1.28)$$

Il sistema, perciò, risulta completamente linearizzato e descritto dalla chain form (riferimento a DescrizioneChainForm):

$$\ddot{z}_1 = u_1 \quad (1.29)$$

$$\ddot{z}_2 = u_2 \quad (1.30)$$

### 1.3 Dimostrazione anolomia del sistema[5]

Considerando  $q = [x, y, \vartheta]$  e, come descritto nel modello cinematico del primo ordine 1.8, si ha che:

$$g_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (1.31)$$

$$g_2 = \begin{bmatrix} \cos\vartheta \\ \sin\vartheta \\ 0 \end{bmatrix} \quad (1.32)$$

$$(1.33)$$

calcolando la  $g_{12}$  si ottiene che:

$$\begin{aligned} g_{12} &= \begin{bmatrix} g_1 & g_2 \end{bmatrix} = \frac{dg_2}{dq} g_1 - \frac{dg_1}{dq} g_2 \\ &= \begin{bmatrix} -\sin\vartheta \\ \cos\vartheta \\ 0 \end{bmatrix} \end{aligned} \quad (1.34)$$

che risulta di rango 1.

Costruendo la matrice  $A = \begin{bmatrix} g_1 & g_2 & g_{12} \end{bmatrix}$  si ottiene:

$$A = \begin{bmatrix} 0 & \cos\vartheta & -\sin\vartheta \\ 0 & \sin\vartheta & \cos\vartheta \\ 1 & 0 & 0 \end{bmatrix} \quad (1.35)$$

che risulta avere  $\det(A) = \cos^2\vartheta + \sin^2\vartheta = 1$  quindi di rango pieno, verificando la completa anolomia del sistema unicyclo considerato.



# Capitolo 2

## Descrizione del khepera

Il Khepera è un robot mobile miniaturizzato con funzionalità simili ad un'ampia gamma di robot utilizzati nei campi didattici e di ricerca. Esso permette test di algoritmi sviluppati in simulazione per trajectory planning, valutazione della posizione di ostacoli (tramite i sensori IR integrati) e molti altri test ed esperimenti.

La disponibilità di librerie già implementate per controllare il robot, per comunicare con un PC e quindi permettere le azioni volute fa sì che il Khepera sia perfetto per la nostra esperienza.

Il robot *Khepera* utilizza un modello differenziale come descritto nel capitolo 1.

### 2.1 Caratteristiche fisiche

Le dimensioni fisiche di particolare interesse sono:

- $r = 8[mm]$  raggio delle ruote
- $L = 53[mm]$  lunghezza asse

### 2.2 Encoder

Su ogni ruota del Khepera è calettato un encoder che misura l'angolo di rotazione compiuto dalla rispettiva ruota. La risoluzione degli encoder è pari a  $100[step/rad]$  comportando un numero totale di step contati in una intera rotazione della ruota pari a  $628step$ , i passi conteggiati data la rotazione  $\phi$  espressa in radianti è pari a:

$$stepGiro = \frac{\phi}{2\pi} 628 \quad (2.1)$$

Essendo la circonferenza delle ruote pari a  $2\pi r_{wheel} = 50,26[mm]$  ad ogni step dell'encoder si ha uno spostamento della ruota pari a:

$$dist = \frac{2\pi r_{wheel}}{stepGiro} = 0,08[mm] \quad (2.2)$$

una precisione molto elevata che permette un buon controllo di posizione del veicolo.

La buona precisione dei risultati ottenuti non ha richiesto la necessità di filtrare i segnali in arrivo dagli encoder.

## 2.3 Sistema di comunicazione e tempi

Il Khepera permette di comunicare sia via cavo che in modalità radio. A causa della scarsa durata delle batterie e del protocollo di comunicazione non ottimizzato in caso di perdita di pacchetti, la scelta è ricaduta sulla comunicazione via cavo, più affidabile, permettendo di valutare in maniera più concreta le prestazioni dei controllori di seguito riportate.

Il periodo minimo di comunicazione (campionamento  $T$ ) dei pacchetti dati è fortemente limitato dal protocollo di comunicazione che richiede circa  $0.02[s]$  per ogni operazione richiesta. Data la necessità di fornire al robot l'azione da compiere e la richiesta della lettura delle posizioni degli encoder, si necessita di almeno  $T > 2 \cdot 0.02[s]$ , a cui vanno ad aggiungersi i tempi di calcolo del controllo che fa sì che il tempo di campionamento minimo si porti a  $T = 0.06[s]$ . Questo dato mette chiaramente in luce un campionamento abbastanza blando, che, come si vedrà in seguito influenzerà particolarmente le prestazioni. I dati presentati sono frutto di una serie di prove effettuate con l'apparecchiatura a disposizione.

## 2.4 Limiti velocità

I limiti di velocità forniti dal data-sheet del robot sono:

- velocità massima  $1000[mm/s]$
- velocità minima  $20[mm/s]$

Dall'analisi da noi eseguita i valori di velocità che ci permettono di lavorare, in funzione anche del campionamento precedentemente presentato, sono:

- velocità massima  $150[mm/s]$  con difficoltà nel controllo
- velocità massima per un buon controllo  $100[mm/s]$

per quanto riguarda il limite minimo di velocità si è valutato che si riesce a scendere fino a  $4[mm/s]$ , soglia limite, al di sotto della quale non si ha movimento a causa degli attriti di primo distacco.

A causa del controllo che verrà effettuato, che si basa sulle letture degli encoder, si è rilevato che una buona velocità per effettuare i test è nell'ordine dei  $50[mm/s]$  che consente di scongiurare il problema dello slip (scivolamento) delle ruote.



# Capitolo 3

## Path Planing

### 3.1 Generazione nel tempo

La generazione di traiettorie nel tempo è la tecnica più facile ed intuitiva e consiste nella generazione di  $x(t)$  e  $y(t)$  in forma analitica o mappata (con punti di passaggio parametrizzati nel tempo). Nel *caso analitico* si ha il calcolo diretto dei valori necessari per la traiettoria, in quanto risulta sufficiente il calcolo delle posizioni al tempo di campionamento prestabilito ( $T$ ).

Nel caso di mappe si possono distinguere due possibili tipologie:

- interpolazione
- campionamento a periodo variabile

La prima tecnica consiste nell'interpolare la traiettoria e determinare la curva di velocità per consentire il rispetto dei tempi, cioè si tratta di *generare le traiettorie nello spazio*, rimandando alla trattazione in 3.2.

La seconda tecnica prevede di *campionare la traiettoria a periodo variabile* che rispetti le specifiche. Questa tecnica presenta maggiori difficoltà di controllo qualora i tempi imposti dal campionamento siano elevati. Per questo motivo, quest'ultima tecnica non è stata implementata, in quanto si tratta di un caso meno generico che introduce maggiori problematiche che porterebbero a delle prestazioni mediocri, a meno di non dedicare uno studio più approfondito che in questa sede non aveva sufficiente tempo a disposizione.

Nel caso di forma analitica si sono prese in considerazione due traiettorie a forma di '8' di diversa dimensione e con velocità di percorrenza diverse, queste sono delle traiettorie di test particolarmente valide in quanto presentano diversi raggi di curvatura e direzione (destra e sinistra) e tratti di bassissima curvatura (approssimabili come rettilinei), ricoprendo uno spettro sufficientemente ampio di tipologie di traiettorie.

Il sistema che genera le traiettorie è:

$$\begin{cases} x(t) = A \sin\left(\frac{t}{B_1}\right) \\ y(t) = A \sin\left(\frac{t}{B_2}\right) \end{cases} \quad (3.1)$$

con  $A = 200$ ,  $B_1 = 5$  e  $B_2 = 10$  nel primo caso (presentato in figura 3.1), e  $A = 1000$ ,  $B_1 = 10$  e  $B_2 = 20$  nel secondo (presentato in figura 3.2).

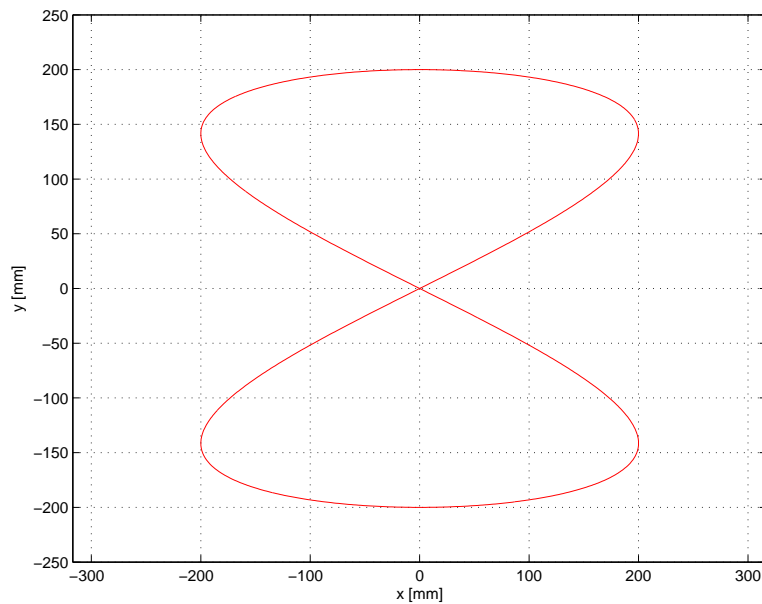


Figura 3.1: Traiettoria ad 8 utilizzata nei test, piccolo

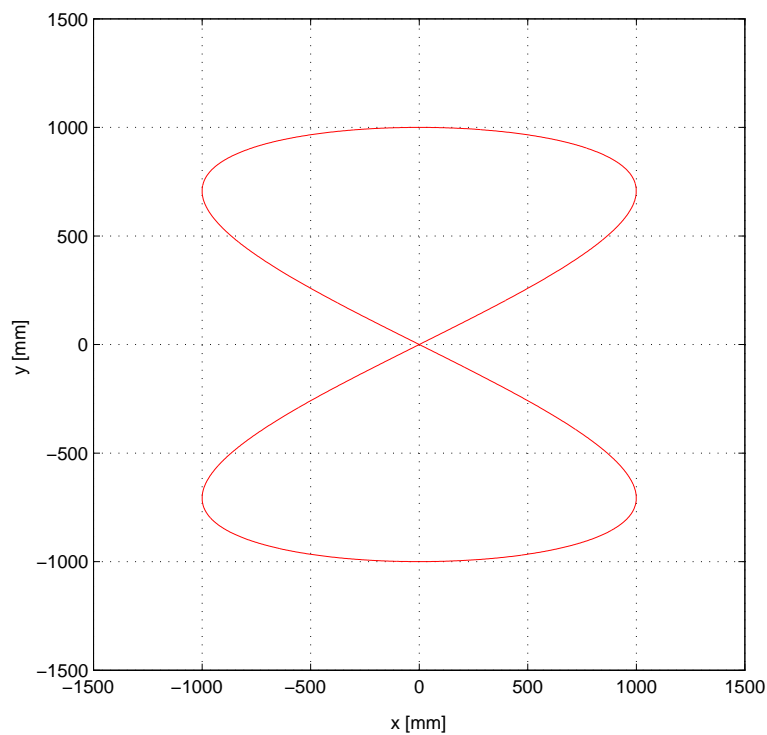


Figura 3.2: Traiettoria ad 8 utilizzata nei test, grande

Sono state utilizzate due traiettorie di uguale forma ma di diversa dimensione

in quanto pongono problematiche di controllo diverse. Infatti, come si vede in figura 3.3 l'8 di minor dimensione viene percorso a velocità notevolmente più basse che in curva mettono in crisi il robot in fase sperimentale, in quanto per valori troppo bassi di velocità le ruote risultano ferme come descritto nel capitolo 2. L'8 di dimensioni maggiori, invece, presenta problematiche legate a velocità sostenute (figura 3.4), problema aggravato dall'elevato tempo di campionamento.

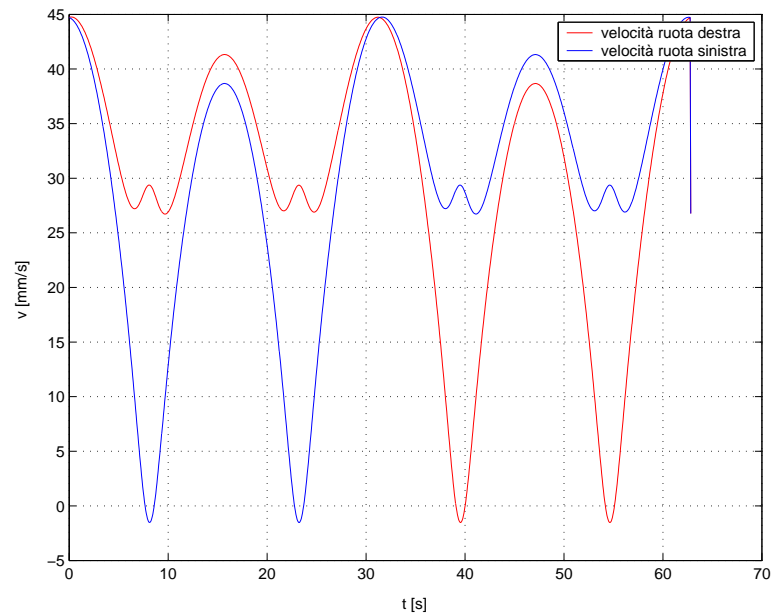


Figura 3.3: Velocità delle ruote ne caso dell'8' piccolo

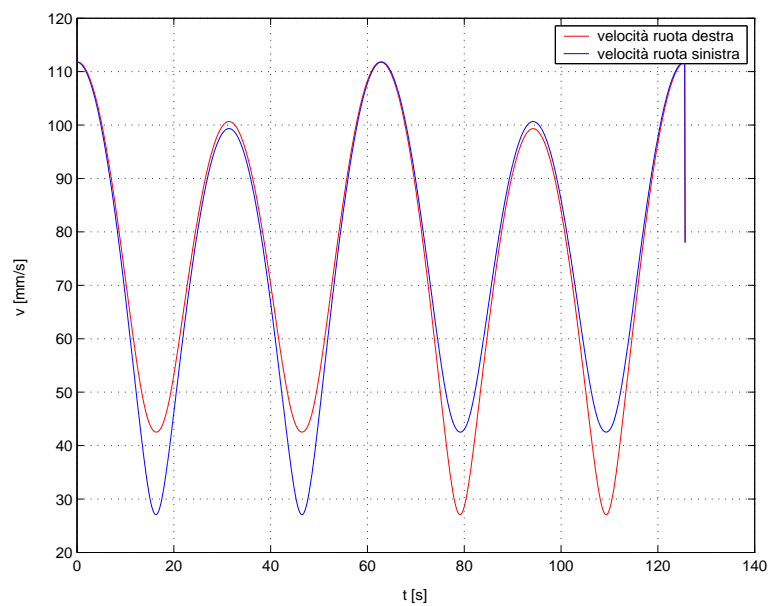


Figura 3.4: Velocità delle ruote ne caso dell'8' grande

### 3.1.1 Sinusoidale (punto-punto)

Il metodo[2] si basa su due fasi consecutive che permettono, tramite ingressi di comando sinusoidali, di portare il veicolo autonomo dalla configurazione iniziale  $q_i = \begin{bmatrix} z_{1i} & z_{2i} & z_{3i} \end{bmatrix}$  a quella finale  $q_f = \begin{bmatrix} z_{1f} & z_{2f} & z_{3f} \end{bmatrix}$  con riferimento al modello a catena cinematico descritto precedentemente.

La prima fase consiste nel portare le variabili  $z_1$  e  $z_2$  dalla posizione iniziale a quella finale ( $z_{1,i} \rightarrow z_{1,f}$  e  $z_{2,i} \rightarrow z_{2,f}$ ) in tempo finito.

Nella seconda fase, per ogni  $z_{k+2}$  con  $k \geq 1$ , si portano le variabili  $z_{k+2}$  al loro valore finale  $z_{k+2,f}$  utilizzando  $v_1 = \alpha \sin \omega t$  e  $v_2 = \beta \cos k \omega t$  in un periodo  $T = 2\pi/\omega$ , con i coefficienti  $\alpha$  e  $\beta$  che soddisfano:

$$\frac{\alpha^k \beta}{k!(2\omega)^k} = z_{k+2,f}(T) - z_{k+2,i}(0) \quad (3.2)$$

che garantisce  $z_i(T) = z_i(0) = z_{i,f}$  per  $i < k$ .

Nel caso in esame  $k$  vale 1, perciò deve essere soddisfatta la condizione:

$$\frac{\alpha \beta}{2\omega} = z_{3,f}(T) - z_{3,i}(0) \quad (3.3)$$

che porta ad avere un coefficiente libero, da scegliere a piacere, con la generazione di diversi risultati che in seguito verranno presentati.

E' possibile eseguire in un colpo solo la seconda fase usando:

$$v_1 = a_0 + a_1 \sin \omega t \quad (3.4)$$

$$v_2 = b_0 + b_1 \cos \omega t + \dots + b_{n-2} \cos(n-2)\omega t \quad (3.5)$$

che nel nostro caso si riduce a

$$v_1 = a_0 + a_1 \sin \omega t \quad (3.6)$$

$$v_2 = b_0 + b_1 \cos \omega t \quad (3.7)$$

Effettuando delle simulazioni con punto iniziale  $q_i = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$  e finale  $q_f = \begin{bmatrix} 200 & 0 & 200 \end{bmatrix}$  in cui  $z_1 = x$ ,  $z_2 = \tan(\vartheta)$  e  $z_3 = y$  e con  $T = 30[s]$  si sono ottenuti i risultati riportati in seguito. Nel calcolo analitico, in realtà, si impone  $T = 1[s]$  per poi scalare opportunamente la velocità di percorrenza e la velocità angolare della traiettoria.

Con queste condizioni, risulta:

$$a_0 = z_{1,f} - z_{1,i} = 200 \quad (3.8)$$

$$b_0 = z_{2,f} - z_{2,i} = 0 \quad (3.9)$$

e scegliendo  $b_1$  come parametro libero che determina il valore di  $a_1$  dalla relazione:

$$a_1 = -\frac{8(2\pi^2(a_0tz_{2,i} - z_{3,f}) + a_0b_1\sin^2(\pi t))}{4b_1\pi t + 16\pi z_{2,i}\sin^2(\pi t) - b_1\sin(4\pi t)} \quad (3.10)$$

Graficamente la relazione risulta riportata in figura 3.5, dove si denota una riduzione di  $a_1$  all'aumentare di  $b_1$ .

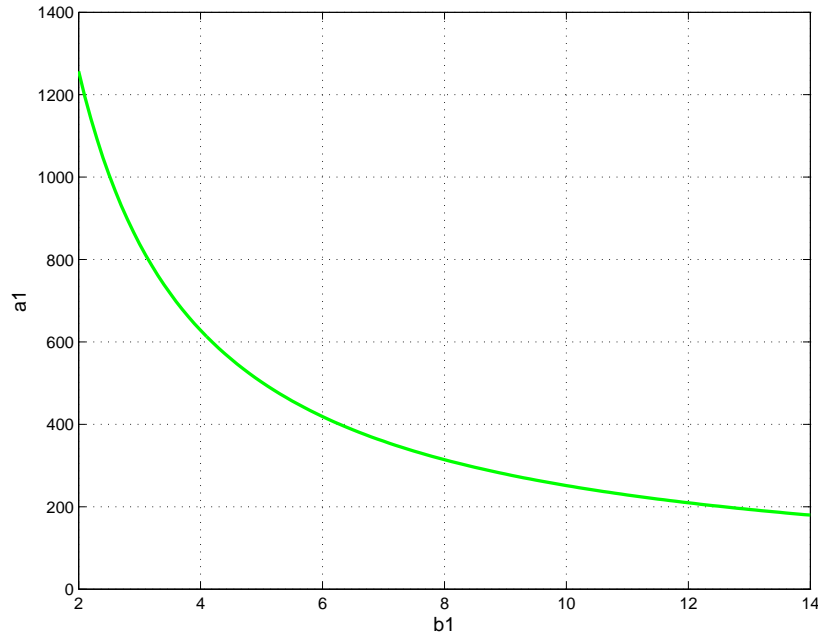


Figura 3.5: Andamento di  $a_1$  in funzione di  $b_1$

Ricordando che aumentare  $b_1$  significa imporre una maggior derivata della tangente di  $\vartheta$  (e per monotonia della funzione tangente anche su  $\vartheta$ ) ci si aspetta che con  $b_1$  piccoli si abbiano elevate variazioni della  $x$  nel tempo e limitate variazioni della  $\vartheta$ . Come si vede in figura 3.6, questo significa, intuitivamente, un'avvicinamento lento lungo l'asse  $y$  con l'insorgere di cuspidi. Sempre in figura si nota che all'aumentare di  $b_1$  (e conseguente diminuzione di  $a_1$ ) aumentano le velocità  $\dot{y}(t)$  e  $\dot{\vartheta}(t)$  e diminuisce  $\dot{x}(t)$ , generando traiettorie con assenza di cuspidi e che raggiungono valori di  $y$  sempre più elevati. Con valore di  $b_1 = 14$  si ottiene una situazione 'ottima' in cui non si ha l'insorgere di cuspidi e si massimizzano le velocità di percorrenza. In figura 3.7 sono presentate le velocità del baricentro in funzione del parametro  $b_1$ , che dimostrano che se la velocità non attraversa l'asse delle ordinate non si ha l'insorgere di cuspidi in quanto non si hanno inversioni delle velocità.

### 3.1.2 Polinomiale (punto-punto)

Un metodo analogo al precedente (presentato anch'esso in [2]), è quello di utilizzare come ingressi di comando degli *input di tipo polinomiale* (con riferimento al sistema a catena cinematico) che risultano da un'evoluzione del caso ad ingressi costanti (denominato 'Steering with piecewise constant input').

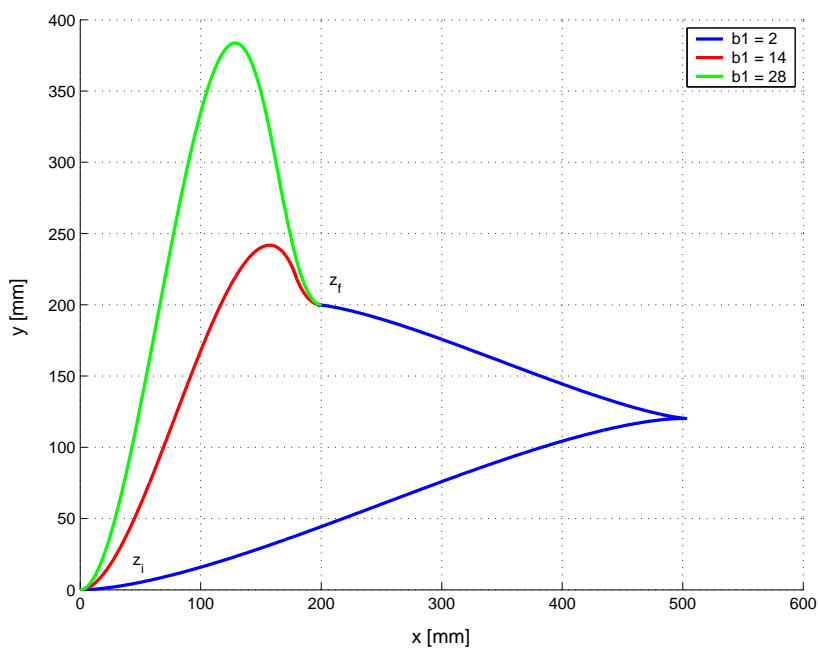


Figura 3.6: Traiettorie ottenute con diverse valori di  $b_1$

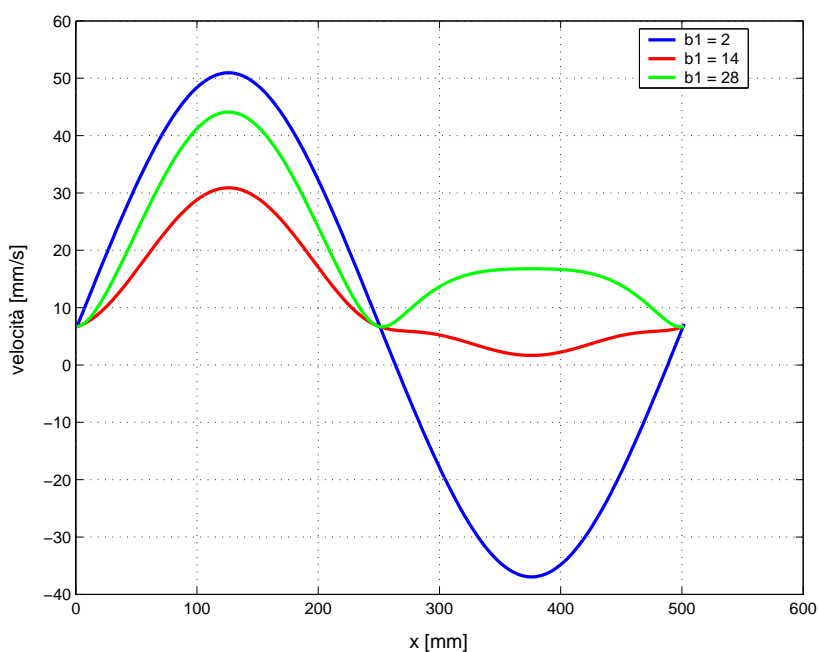


Figura 3.7: Velocità del baricentro in funzione di  $b_1$

Il controllo viene eseguito utilizzando:

$$v_1 = \text{sign}(z_{1,f} - z_{1,i}) \quad (3.11)$$

$$v_2 = c_0 + c_1 t + \dots + c_{n-2} t^{n-2} \quad (3.12)$$

con  $T = z_{1,f} - z_{1,i}$  e  $c_0, \dots, c_{n-2}$  ottenuti risolvendo il sistema lineare incominciando dall'integrazione della forma chiusa del modello di equazioni:

$$M(T) \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-2} \end{bmatrix} + m(z_i, T) = \begin{bmatrix} z_{2,f} \\ z_{3,f} \\ \vdots \\ z_{n,f} \end{bmatrix} \quad (3.13)$$

con  $M(T)$  matrice non singolare per  $T \neq 0$ .

Essendo  $T$  determinato da  $z_{1,f} - z_{1,i}$  anche in questo caso è necessario applicare una riscalatura in funzione del tempo totale di percorrenza che si vuole imporre.

Nel caso in esame il sistema si riduce alla determinazione delle costanti  $c_0$  e  $c_1$  in quanto gli ingressi risultano:

$$v_1 = \text{sign}(z_{1,f} - z_{1,i}) \quad (3.14)$$

$$v_2 = c_0 + c_1 t \quad (3.15)$$

essendo il sistema anolonomo descritto da sole 3 variabili di stato.

Casi particolari si possono identificare qualora avvenga una delle seguenti condizioni:

1.  $z_{1,f} = z_{1,i}$
2.  $T$  molto piccolo

Per quanto riguarda la prima è facilmente risolvibile ponendo un punto di passaggio intermedio e sviluppando l'analisi su due distinte curve concatenate, il secondo, invece, porta a ingressi di comando molto ampi che rischiano di portare gli attuatori a saturazione, rendendo le prestazioni ideali non verificate in sperimentazione. Per quest'ultimo motivo si può operare come precedentemente detto imponendo il tempo  $T$  dell'azione ad un valore sufficientemente elevato da scongiurare l'insorgere di saturazioni.

La traiettoria ottenuta nelle medesime condizioni utilizzate con ingressi sinusoidali, riportata in figura 3.8, è chiaramente più efficiente, ed oltre a non presentare nessun tipo di cuspidi, le velocità necessarie (presentate in figura 3.9) sono molto più basse.

### 3.1.3 Limiti alla generazione di traiettorie nel tempo

Nel caso di generazione di traiettorie con le tecniche descritte nei paragrafi 3.1.1 e 3.1.2 vi è la possibilità che la velocità vada a zero, il che pone dei limiti alla tipologia di controllo che si vuole implementare. In questo caso è possibile intraprendere due diverse strade, la prima più semplice ed immediata che si basa sul sommare, qualora la velocità si annulli, un  $\epsilon > 0$  che permetta di risolvere il problema. La seconda soluzione, più elaborata e corretta, si basa sul fatto di

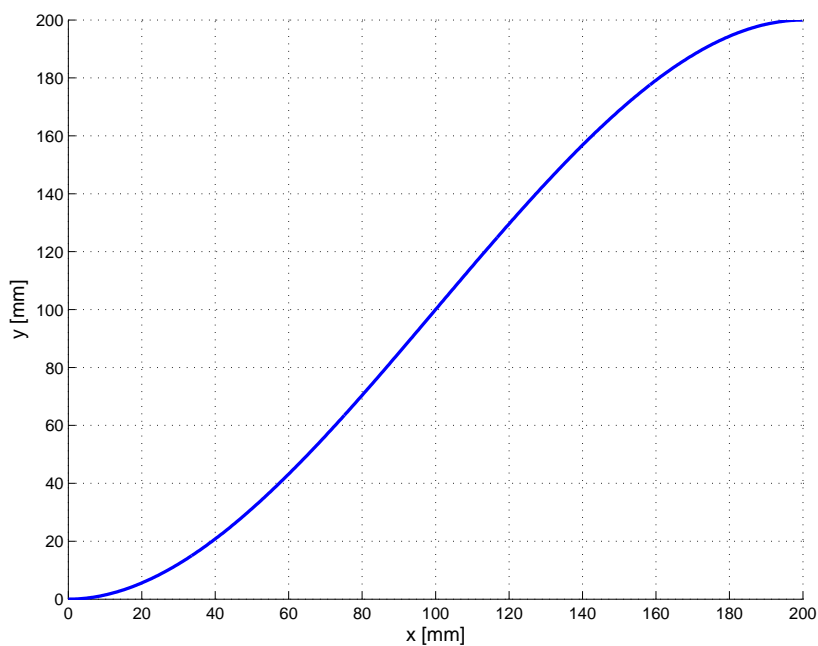


Figura 3.8: Traiettoria ottenuta con il metodo polinomiale

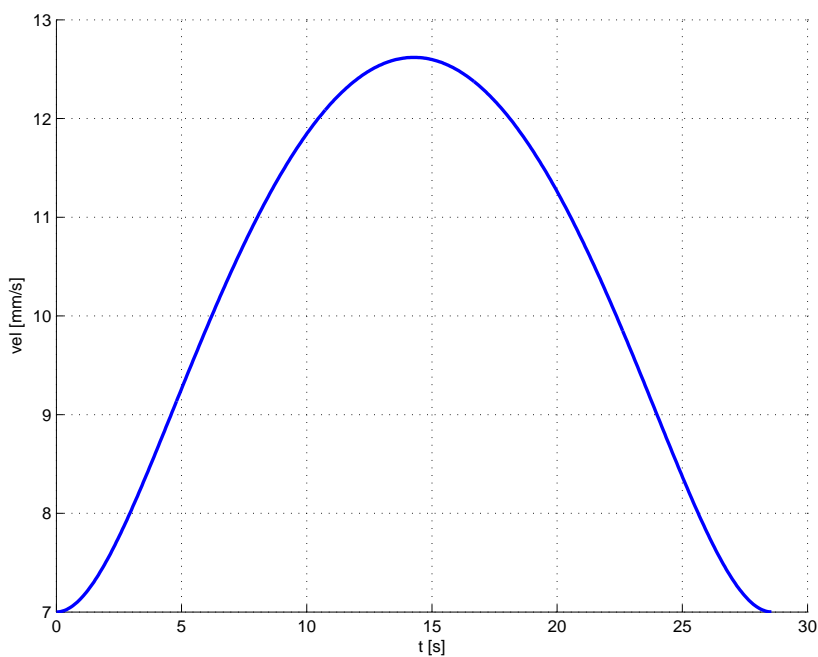


Figura 3.9: Profilo di velocità nel caso polinomiale

scindere in due la traiettoria calcolata proprio sul punto a velocità nulla (che generalmente coincide con il punto di inversione della velocità) e calcolare separatamente le due traiettorie. Quest'ultimo risulta particolarmente impegnativo qualora vi siano  $m$  cambi di direzione, comportando la rottura della traiettoria in  $m + 1$  sottotraiettorie distinte.



La parametrizzazione nel tempo non permette di garantire il passaggio per punti prestabiliti impedendo la gestione di ostacoli intermedi o limiti sul piano di lavoro.

Volendo ridurre le velocità (tramite gestione del parametro libero) si ampliano gli spazi percorsi in quanto le manovre richiedono angoli di curvatura più elevati, portando, di conseguenza a tempi di percorrenza elevati. In modo opposto, le velocità possono essere imposte più elevate, con la generazione di traiettorie più ristrette e veloci (in termini di tempo totale) di percorrenza, ma con la generazione di segnali di comando particolarmente elevati con conseguente saturazione degli attuatori.

Risulta, inoltre, impossibile gestire una curva di velocità, condizione che molto spesso risulta utile per percorrere tracciati misti.

## 3.2 Generazione nello spazio

Quanto detto sinora non permette né una gestione spaziale delle traiettorie né un'imposizione libera di curve di velocità su di esse. Per questo motivo è necessario implementare un algoritmo che data una traiettoria permetta di ricavare la  $\vartheta(t)$  a partire da una curva di velocità. Nel caso si voglia imporre il passaggio per più punti, la traiettoria viene generata tramite spline polinomiali e su di essa viene applicato tale algoritmo.

### 3.2.1 Imposizione di velocità su traiettoria analitica

Nel caso in cui si abbia a che fare con traiettorie analitiche, del tipo  $y = f(x)$ , e si voglia imporre una curva di velocità si può operare ricavando la lunghezza della traiettoria tramite l'integrazione:

$$\int_0^x \sqrt{1 + \left(\frac{df(x)}{dt}\right)^2} dx \quad (3.16)$$

e, ricordando che si lavora in forma campionata, risolvere numericamente l'equazione:

$$kT = \int_0^x \sqrt{1 + \left(\frac{df(x)}{dt}\right)^2} dx \quad k = 0, 1, \dots \quad (3.17)$$

dove  $T$  è il tempo di campionamento. Dal risultato ottenuto si calcolano, banalmente,  $y(t) = f(x(t))$  e  $\omega(t) = \dot{\vartheta}(t) = \frac{d}{dt} \left[ \arctg \left( \frac{y(t)}{x(t)} \right) \right]$  riconducendosi al caso precedentemente descritto di traiettorie nel tempo.

Successivamente si verificherà la correttezza di tale calcolo con delle simulazioni.

### 3.2.2 Spline

Nel caso in cui la traiettoria di percorrenza non sia data in forma analitica ma si abbiano solamente i punti di passaggio è necessario, prima di effettuare l'integra-

zione della lunghezza della curva, determinare quest'ultima. Una possibile scelta è l'impiego di spline polinomiali.

Il loro utilizzo permette di calcolare agevolmente polinomi di diverso grado che incrocino i punti di passaggio forniti eventualmente con imposizioni sulle loro derivate. Per comodità nelle successive simulazioni si sono utilizzate *spline cubiche* che garantiscono solo il passaggio per i punti forniti, senza vincoli sulle derivate.

In questo modo si ottengono  $N - 1$  polinomi di ordine prefissato, con  $N \geq 2$  il numero di punti arbitrari di passaggio richiesti.

A titolo di esempio si sono utilizzati i seguenti punti di passaggio:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 & 100 & 200 & 300 & 400 & 500 & 600 & 700 & 800 \\ 0 & -101.3 & -174.6 & -199.9 & -170.2 & -93.6 & 8.8 & 108.8 & 178.8 \end{bmatrix}$$

ottenuti discretizzando la sinusoida  $y(x) = 200\sin(x)$  con passo 100. La traiettoria che si ottiene è presentata in figura 3.10.

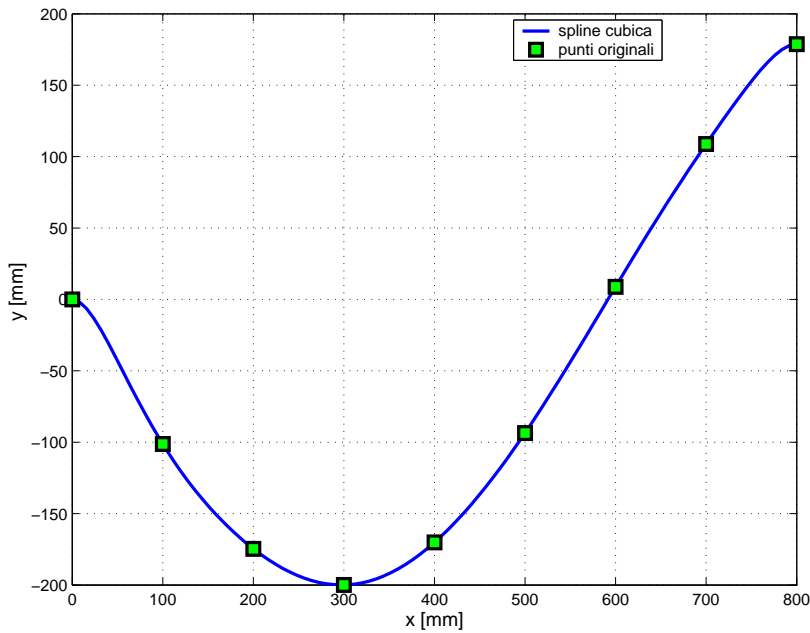
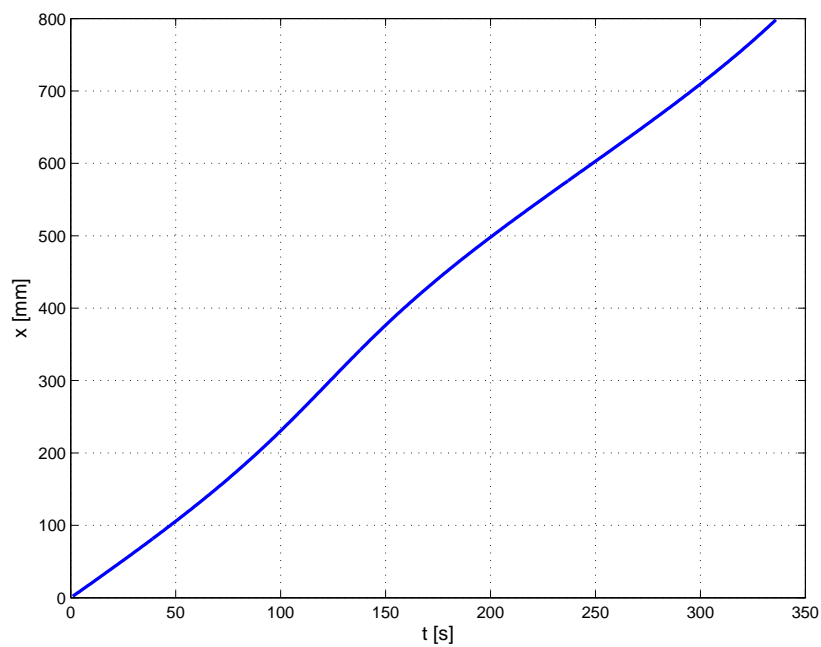
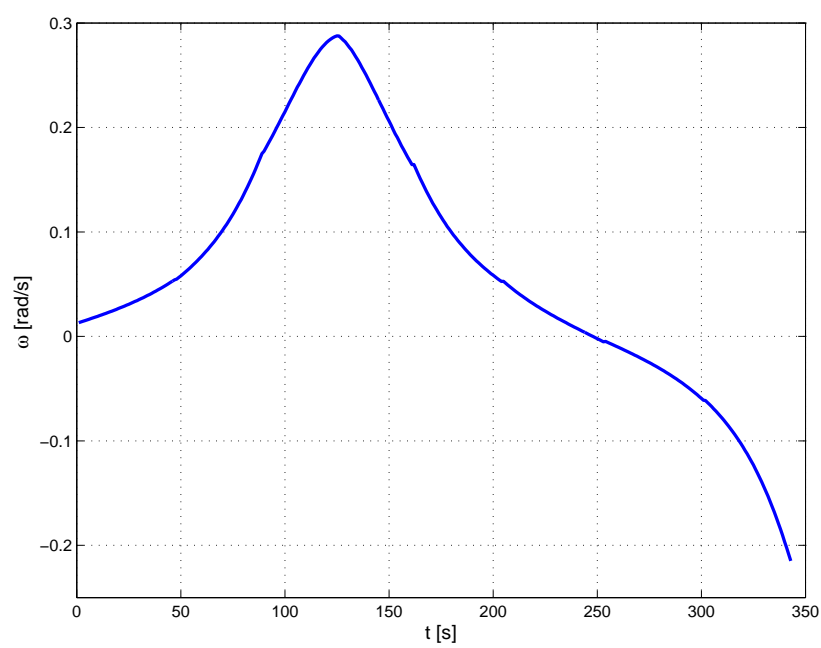


Figura 3.10: Esempio di spline cubica interpolante

Si è scelto di percorrere la traiettoria a velocità costante, che risulta essere una condizione critica in quanto le fasi di curvatura vengono percorse alla stessa velocità dei rettilinei. In più, ciò permette una risoluzione più semplice dell'equazione 3.17.

L'andamento di  $x(t)$  e di  $\vartheta(t)$  sono riportate rispettivamente in figura 3.11 e 3.12.

Figura 3.11: Andamento di  $x(t)$ Figura 3.12: Andamento di  $\vartheta(t)$



# Capitolo 4

## Azioni di controllo

La fase successiva al calcolo della posizione  $(x(t), y(t))$  e della rispettiva velocità angolare del baricentro  $\omega(t)$  come descritto nel capitolo 3, è quella di compiere l'azione di controllo sul robot, presentata sia in catena aperta (feed-forward) che con diverse tipologie di controllo in catena chiusa.

### 4.1 Generazione azione feed-forward

L'azione di feed-forward si basa sulla generazione degli ingressi necessari al sistema per eseguire l'operazione richiesta, che nel caso in esame sono le velocità da imporre alle ruote, facilmente ricavabili dalla velocità del baricentro e dalla velocità angolare. Questa tecnica, in catena aperta, ha notevoli limitazioni in campo sperimentale, ma è un ottimo complemento qualora accoppiata ad un controllore *PD* come vedremo in seguito.

Le relazioni risultano quindi:

$$\begin{cases} v_L = \left(v + \omega \frac{L}{2}\right) \\ v_R = \left(v - \omega \frac{L}{2}\right) \end{cases} \quad (4.1)$$

ottenendo in maniera semplice ed immediata i comandi da fornire alle ruote del Khepera.

### 4.2 Controllori implementati

Come precedentemente trattato, la singola azione del feed-forward non è sufficiente a garantire un buon controllo in catena aperta, è più conveniente affiancare a quest'azione un controllore che corregga l'errore che si presenta, andando a completare il comando.

Per fare ciò sono stati implementate 3 tipologie di controllo denominate 'linear desing', 'nonlinear desing' e 'Dynamic feedback linearization' descritte in seguito e presentate correlate dalle simulazioni compiute.

### 4.2.1 Linear desing

Effettuando un preliminare cambio di variabili nel modello dell'uniciclo, successivamente linearizzando l'errore dinamico lungo la traiettoria desiderata, ed in conclusione imponendo il loop chiuso desiderato (con un desing lineare di controllo degli input), si ottengono le seguenti equazioni:

$$v = v_d \cos(\vartheta_d - \vartheta) + k_1 [(x_d - x) \cos\vartheta + (y_d - y) \sin\vartheta] \quad (4.2)$$

$$\omega = \omega_d + k_2 \text{sign}(v_d) \cdot [(y_d - y) \cos\vartheta - (x_d - x) \sin\vartheta] + k_3 (\vartheta_d - \vartheta) \quad (4.3)$$

Una scelta consigliata dei parametri del controllo è suggerita in [?] e vale:

$$k_1 = k_3 = 2\zeta \sqrt{\omega_d^2(t) + bv_d^2(t)} \quad k_2 = b |v_d^2(t)| \quad (4.4)$$

con  $\zeta \in (0, 1)$  e  $b > 0$ .

### 4.2.2 Nonlinear desing

Un altro metodo di tracking control si basa su una appropriata funzione di Lyapunov[?]:

$$v = v_d \cos(\vartheta_d - \vartheta) + k_1 [(x_d - x) \cos\vartheta + (y_d - y) \sin\vartheta] \quad (4.5)$$

$$\omega = \omega_d + k_2 v_d \frac{\sin(\vartheta_d - \vartheta)}{\vartheta_d - \vartheta} [(y_d - y) \cos\vartheta - (x_d - x) \sin\vartheta] + k_3 (\vartheta_d - \vartheta) \quad (4.6)$$

In modo analogo al caso precedente la scelta dei parametri è:

$$k_1 = k_3 = 2\zeta \sqrt{\omega_d^2(t) + bv_d^2(t)} \quad k_2 = b \quad (4.7)$$

### 4.2.3 Dynamic feedback linearization

L'ultima tipologia di controllo presentata si basa sul modello dinamico presentato in 1.2.

Assumendo che il robot debba seguire la traiettoria  $(x_d(t), y_d(t))$ , considerata *persistente*, cioè in modo da verificare che  $v_d = (\dot{x}_d^2 + \dot{y}_d^2)^{1/2}$  lungo la traiettoria.

A partire dal sistema equivalente e disaccoppiato:

$$\ddot{z}_1 = u_1 \quad (4.8)$$

$$\ddot{z}_2 = u_2 \quad (4.9)$$

è immediato progettare una retroazione stabilizzante esponenzialmente per la traiettoria desiderata con:

$$u_1 = \ddot{x}_d + k_{p,1}(x_d - x) + k_{d,1}(\dot{x}_d - \dot{x}) \quad (4.10)$$

$$u_2 = \ddot{y}_d + k_{p,2}(y_d - y) + k_{d,2}(\dot{y}_d - \dot{y}) \quad (4.11)$$

con il controllore proporzionale-derivativo (PD) con guadagni  $k_{j,i} > 0$  con  $j = p, d$  e  $i = 1, 2$ .

Questi segnali, applicati al compensatore dinamico:

$$\dot{\xi} = u_1 \cos \vartheta + u_2 \sin \vartheta \quad (4.12)$$

$$v = \xi \quad (4.13)$$

$$\omega = \frac{u_2 \cos \vartheta - u_1 \sin \vartheta}{\xi} \quad (4.14)$$

permettono di ottenere gli ingressi di controllo  $(v, \omega)$ .





# Capitolo 5

## Simulazioni

Come primo approccio alla fase di controllo dell'uniciclo si è operato in simulazione implementando le strategie di controllo descritte nel precedente capitolo.

In questo capitolo vengono presentati i risultati ottenuti in simulazione sulle due traiettorie test descritte nel capitolo 3 (path ad '8'). Per completare verranno presentate anche le velocità di percorrenza delle traiettorie e della rispettiva  $\omega(t)$ .

### 5.1 Modelli del sistema

Per la simulazione del sistema autonomo è necessario utilizzare due sistemi lineari consecutivi, ricavando  $\vartheta$  e successivamente elaborando le posizioni  $(x, y)$ . Ciò può essere fatto sia in tempo continuo che in tempo discreto. Nei successivi paragrafi verranno presentate in dettaglio le modellizzazioni utilizzate.

#### 5.1.1 Modello continuo

Nel caso a tempo continuo si ha che:

$$\omega(t) = \dot{\vartheta}(t) = \frac{v_R(t) - v_L(t)}{L} \quad (5.1)$$

Calcolata la velocità angolare di rotazione intorno al baricentro è immediato ricavare le velocità lungo gli assi di riferimento del fixed frame:

$$\begin{aligned} \dot{x}(t) = v_x(t) &= \frac{v_R(t) + v_L(t)}{2} \cos(\vartheta(t)) \\ &= v(t) \cos(\vartheta(t)) \end{aligned} \quad (5.2)$$

$$\begin{aligned} \dot{y}(t) = v_y(t) &= \frac{v_R(t) + v_L(t)}{2} \sin(\vartheta(t)) \\ &= v(t) \sin(\vartheta(t)) \end{aligned} \quad (5.3)$$

ed integrando quest'ultime si ottiene la posizione nel piano  $(x(t), y(t))$ .

### 5.1.2 Modello discreto

Nel modello discreto, i passi compiuti sono analoghi e si ha che la velocità angolare vale:

$$\vartheta((k+1)T) = \frac{v_R(kT) - v_L(kT)}{L}T + \vartheta(kT) \quad (5.4)$$

e di conseguenza le posizioni del sistema valgono:

$$x((k+1)T) = x(kT) + v(k)\cos(\vartheta(kT))T \quad (5.5)$$

$$y((k+1)T) = y(kT) + v(k)\sin(\vartheta(kT))T \quad (5.6)$$

Il vantaggio presentato dal modello discreto è di permettere il calcolo on-line del passo successivo senza richiedere integrazioni. Inoltre, per introdurre in simulazione gli effetti dovuti al campionamento, quest'ultimo modello risulta preferibile.

## 5.2 Azione Feed-Forward

Il risultato della sola azione feed forward nel caso dell''8' più grande non è particolarmente apprezzabile dalla traiettoria presentata in figura 5.1, perciò faremo riferimento all'andamento degli errori di figura 5.2.

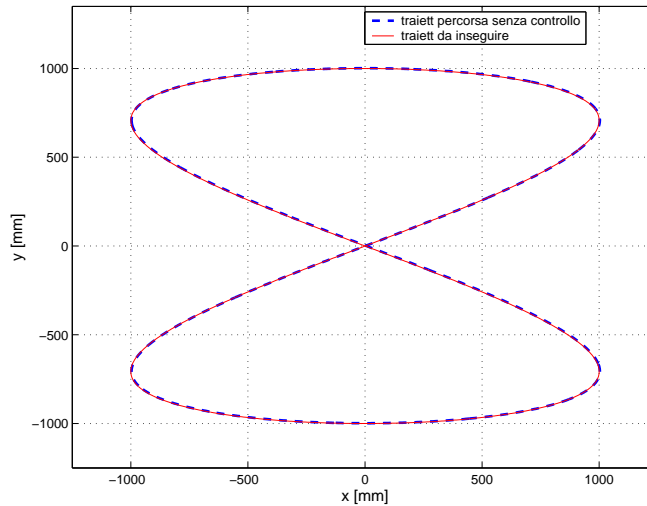


Figura 5.1: Risultato azione feed forward

Per lo stesso motivo presentiamo solo l'errore commesso dal feed forward nel caso dell''8' piccolo in figura 5.3.

Si nota chiaramente che l'errore è direttamente legato alla tipologia di traiettoria ed in particolare, poichè siamo in simulazione, ha origine dall'utilizzo di un modello discreto per simulare il sistema reale. Come si può notare, infatti,

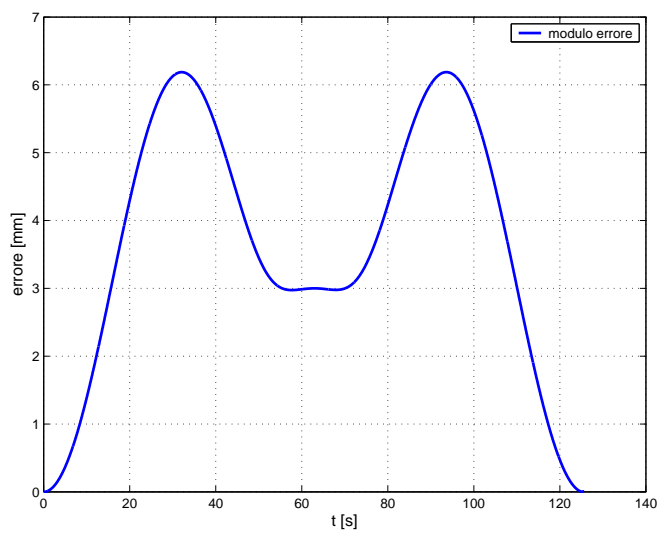


Figura 5.2: Errore compiuto con azione feed forward su 8 grande

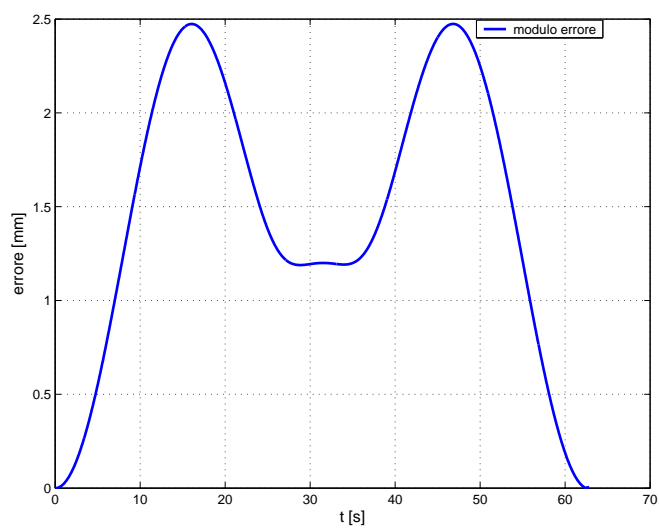


Figura 5.3: Errore compiuto con azione feed forward su 8 piccolo

maggiori sono le velocità, maggiore è l'errore commesso. I punti critici della traiettoria, dove il modulo dell'errore è maggiore, sono quelli orizzontali di raccordo tra le curve.

I risultati presentati sarebbero soddisfacenti, ma essendo in simulazione, la sola azione in catena aperta è insufficiente a compensare errori e scostamenti del sistema reale. Si giustifica quindi lo sviluppo di un'azione di controllo in retroazione.

## 5.3 Azioni di controllo

Nei sottoparagrafi seguenti vengono presentati i risultati ottenuti in simulazione utilizzando azioni di controllo in catena chiusa, sempre rispetto alle due traiettorie test. Poiché queste traiettorie test non tengono in considerazione tutte le possibili condizioni (cuspidi) nel paragrafo successivo verrà trattata separatamente questa particolare condizione.

### 5.3.1 Linear design

Eseguendo un'analisi sull'errore massimo ottenuto in simulazione (lungo la traiettoria a 8 piccolo) con il controllo Linear Design si ottiene il grafico di figura 5.4 in cui si è presentato l'errore massimo compiuto nell'inseguimento in funzione dei parametri  $\zeta$  e  $b$ . Si nota come solo nell'area  $0 < b < 0.2$  e  $0.1 < \zeta < 0.8$  si ottengono errori limitati nell'ordine dei  $2 - 3[mm]$ , e quindi un controllo soddisfacente, nel caso si incrementi eccessivamente  $b$  l'errore sale velocemente, a maggior ragione se in contemporanea si incrementa anche  $\zeta$  che provoca un comportamento esplosivo, che nel grafico viene limitato a  $50[mm]$  solo per motivi grafici, in realtà si ha la perdita di controllo del sistema.

Il grafico ci dà una visualizzazione intuitiva dell'errore del controllo al variare dei due parametri, che di per se non avendo un effetto intuitivo sul sistema controllato sono di difficile taratura. I valori così calcolati sono una base di partenza per effettuare il tuning nelle prove sperimentali.

In fase di simulazione, utilizzando i valori ottimi così trovati ( $b = 0.12$  e  $\zeta = 0.9$ ), si ottiene un andamento dell'errore riportato in figura 5.5. Si determina un errore inferiore ai 2 decimi di millimetro che rende inutile la presentazione grafica della traiettoria percorsa a confronto con quella desiderata.

A completamento, per questa prima tipologia di controllo, si riportano in figura 5.6 l'andamento dell'errore nel caso dell'8 grande. Come si vede, l'errore è molto simile a quello della traiettoria piccola, ovviamente ottenuto ritardando il controllo con valori di  $b = 0.025$  e  $\zeta = 0.9$ . Nelle analisi dei prossimi controllori non verrà più riportato il risultato di quest'ultima traiettoria in quanto non presenta situazioni d'interesse.

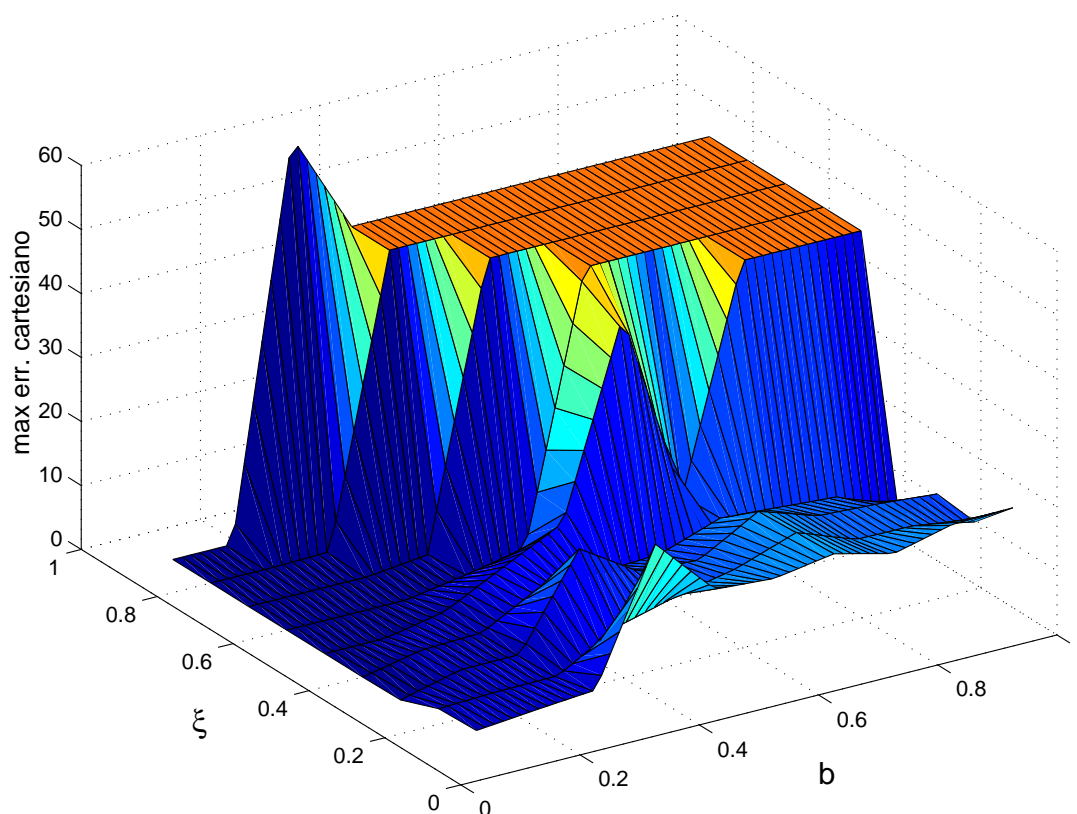


Figura 5.4: Errore massimo ottenuto in sperimentazione con Linear Design, in funzione di  $\zeta$  e  $b$ , su 8 piccolo

### 5.3.2 Non linear design

In modo analogo al caso precedente sono presentati i risultati dell'errore massimo di inseguimento in funzione dei parametri di controllo  $\zeta$  e  $b$  in figura 5.7. Esso mette in luce l'area in cui l'errore di inseguimento è limitato nell'intorno dei  $2 - 3[mm]$  ed è data da valori di  $\zeta$  compresi in  $[0.5, 0.9]$  e  $b$  in  $(0, 0.25]$ .

In figura 5.8 si ottengono risultati analoghi al caso precedente, l'errore è molto piccolo mettendo in luce una certa analogia con il Linear Design. I parametri utilizzati nel controllo sono  $b = 0.12$  e  $\zeta = 0.9$ .

### 5.3.3 Dinamic feedback linearization

In questo caso è necessario effettuare il tuning di entrambi i controllori PD, ma per simmetria si è applicato un controllo parallelo ponendo  $k_{p,1} = k_{p,2}$  e  $k_{d,1} = k_{d,2}$ . Ciò che risulta dall'analisi dell'errore massimo nell'inseguimento dell'8 piccolo è riportato in figura 5.9 dove si denota chiaramente un'aumento dei valori rispetto ai due controlli precedenti. L'area ottima in cui eseguire il tuning manuale in fase sperimentale è quella che si ottiene con  $1.5 < k_p < 4$  e  $k_d > 0.3$ .

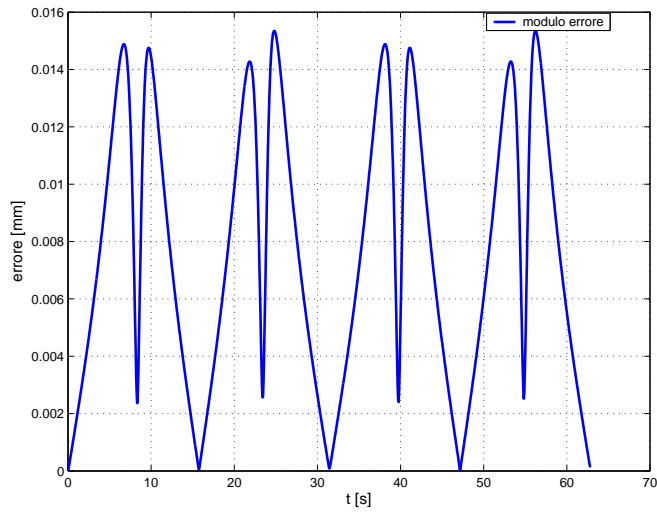


Figura 5.5: Errore con controllo Linear Design su 8 piccolo

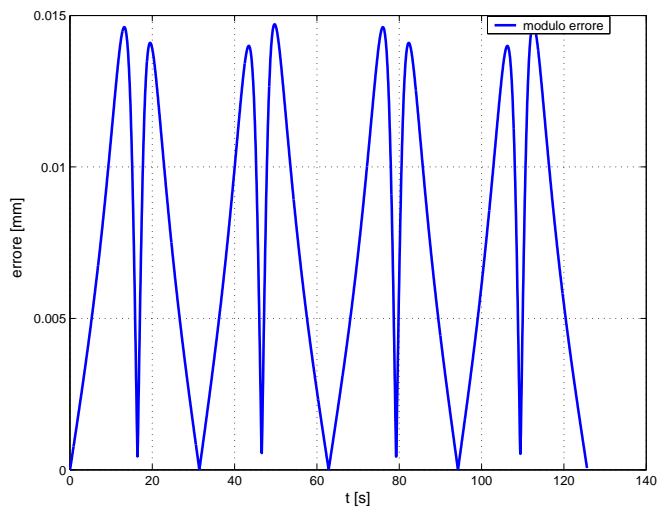


Figura 5.6: Errore con controllo Linear Design su 8 grande

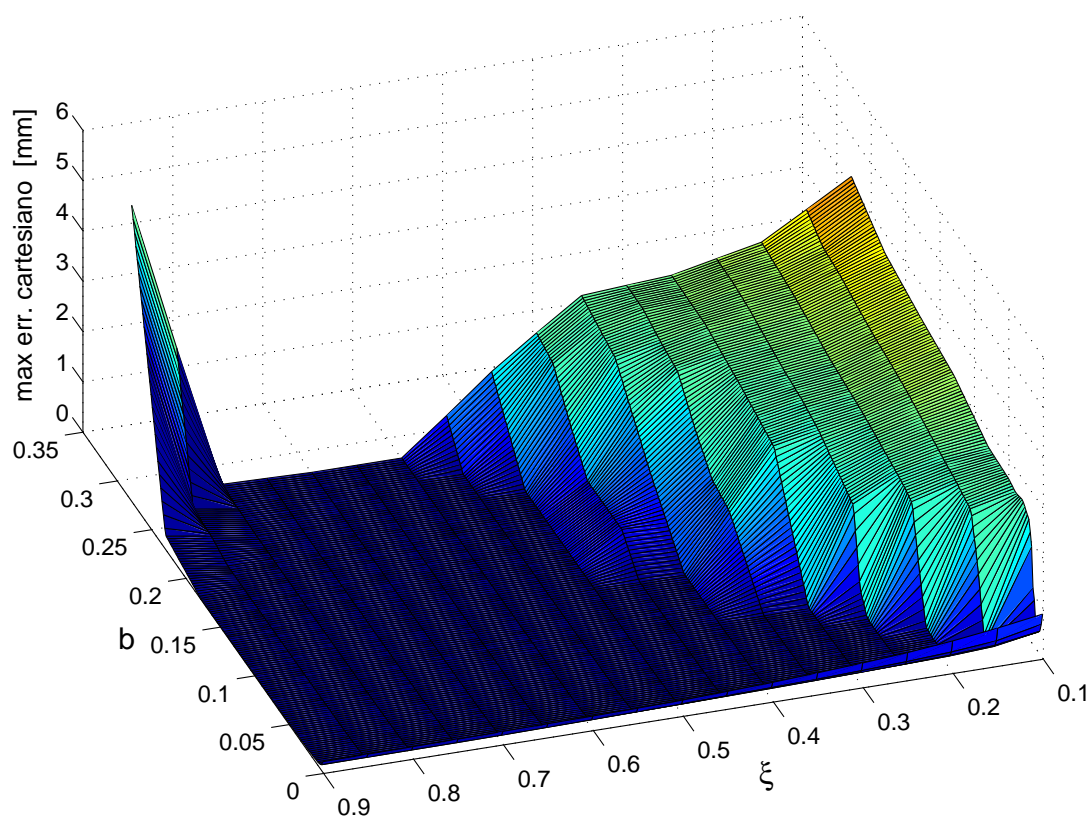


Figura 5.7: Errore massimo ottenuto in sperimentazione con Nonlinear Design, in funzione di  $\zeta$  e  $b$ , su 8 piccolo

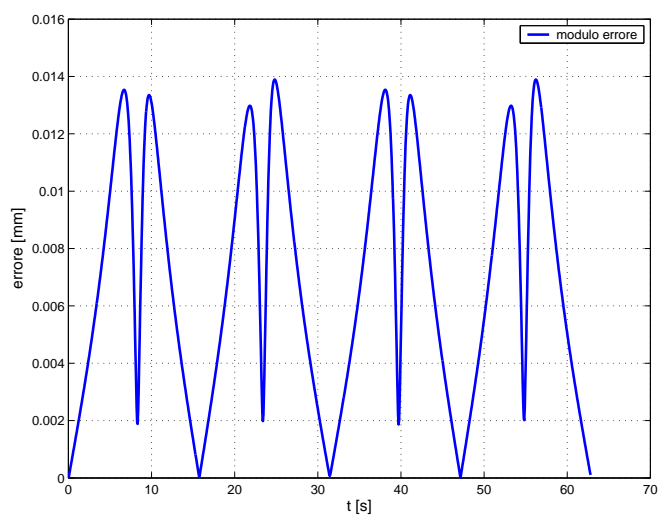


Figura 5.8: Errore con controllo Nonlinear Design su 8 piccolo

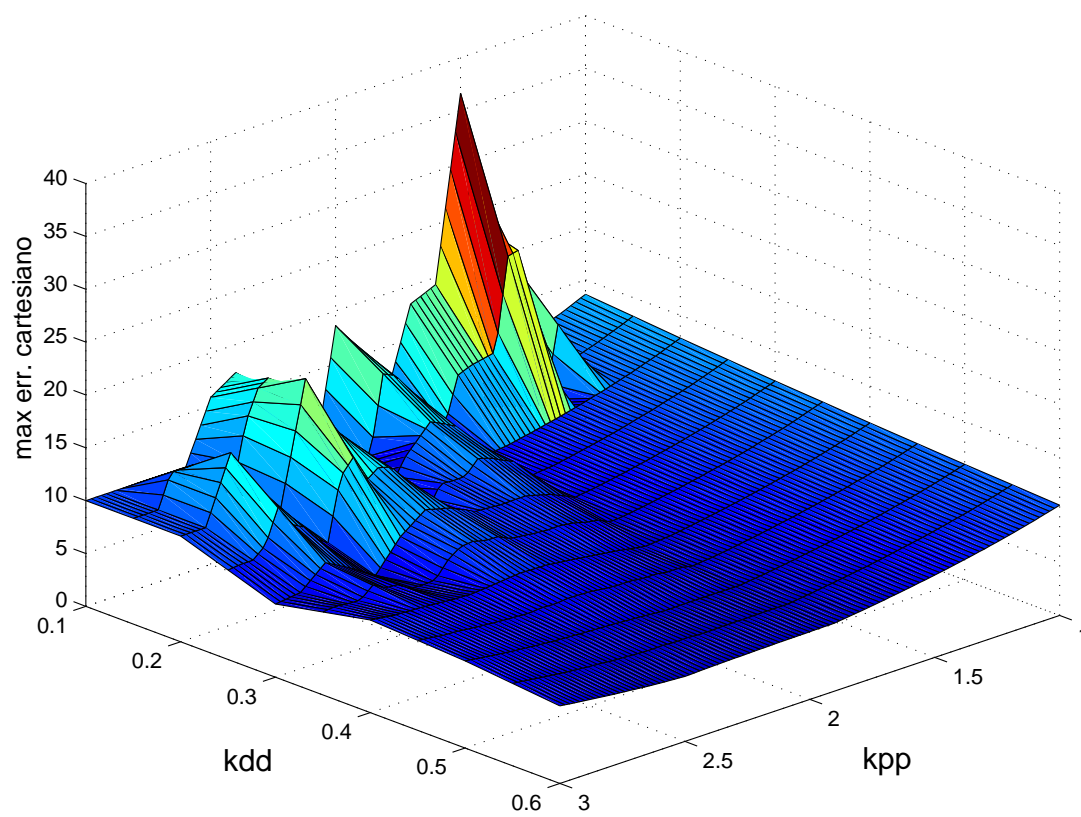


Figura 5.9: Errore massimo ottenuto in sperimentazione con Dinamic Feedback Linearization, in funzione di  $k_{pp}$  e  $k_{dd}$ , su 8 piccolo



Si assumono come valori ottimi  $k_p = 3$  e  $k_d = 3.1$ , con cui si ottiene l'errore riportato in figura 5.10. Il risultato ottenuto è insoddisfacente se confrontato con i risultati precedenti. La spiegazione di ciò non è di facile da trovare in quanto rispetto ai precedenti controllori si sfrutta un modello dinamico, che necessita di un'integrazione ed una derivazione in più.

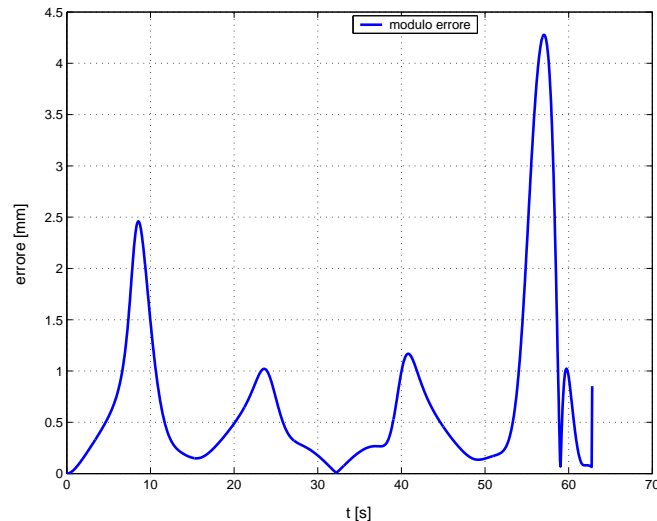


Figura 5.10: Errore con controllo Dinamic Feedback Linearization su 8 piccolo

## 5.4 Limiti del controllo

Nel caso ci si voglia spingere ad ottenere prestazioni sempre più elevate si ottiene un effetto di oscillazione analogo ad un proporzionale elevato nel PID (aggravato dalla presenza di un campionamento poco fitto). In figura 5.11 si osserva, a titolo esemplificativo, cosa accade se con il Nonlinear Design si aumenta il valore di  $b$ .

In figura 5.12 viene riportata l'ultima verifica della bontà dei controllori. Si è applicato un'angolo di partenza nullo, diverso dall'angolo reale (peri a circa  $26,5[gradi]$ ), e verificato che il controllo riesce a sopperire alla mancanza d'informazioni iniziali.

## 5.5 Cuspidi

Infine si presentano i risultati relativi alla traiettoria, con presenza di cuspidi, generate con ingressi sinusoidali sul sistema a catena. In questo caso possono presentarsi problemi di controllo, in particolare per il Dinamic Feedback Linearization. Infatti i primi 2 controllori sviluppati permettono senza nessun tipo di modifica un controllo assolutamente soddisfacente. Nelle figure 5.14 e 5.16 è riportata in dettaglio l'inseguimento di una cuspidine con e senza azione di controllo (nelle immagini 5.13 e 5.17 è illustrato l'inseguimento in generale).

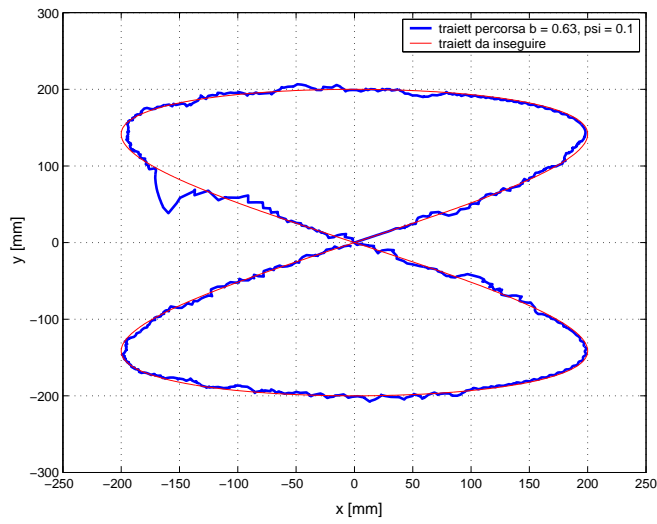


Figura 5.11: Oscillazione sull'8 piccolo con Nonlinear Design su 8 piccolo

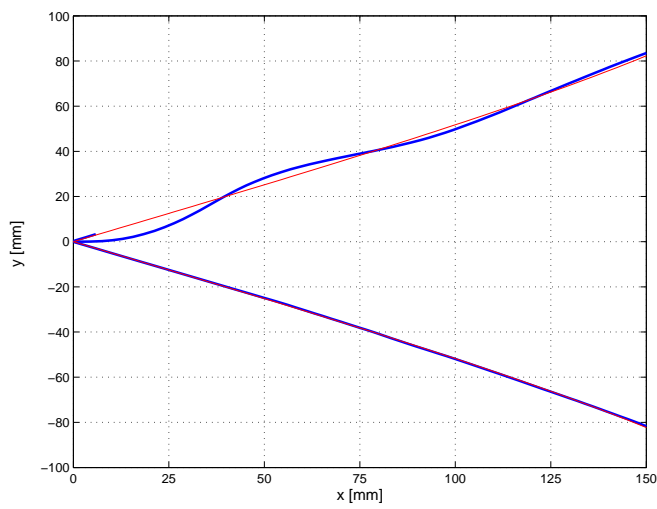


Figura 5.12: Inseguimento con  $\vartheta_0 = 0$

Con il terzo controllore non si è riusciti ad ottenere un controllo soddisfacente in quanto la  $v(t)$  si presenta a denominatore nel calcolo di  $\omega(t)$  secondo la formula 1.24 creando un'esplosione in quest'ultima quando la velocità si annulla. Adattare questo tipo controllo resta un problema aperto.

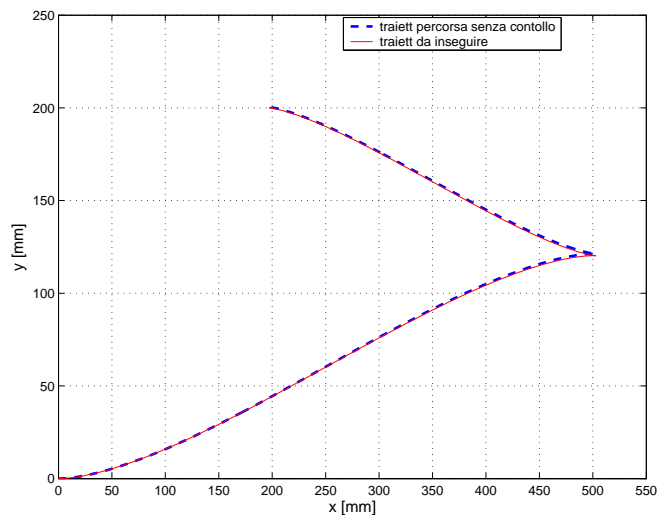


Figura 5.13: Inseguimento traiettoria con cuspidi (solo feed forward)

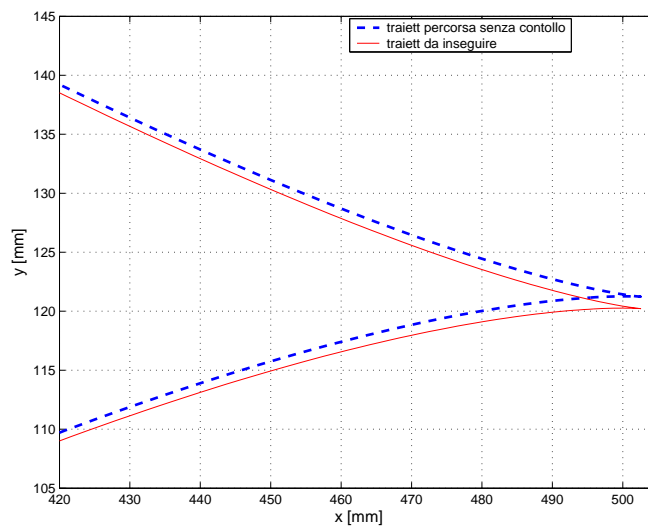


Figura 5.14: Particolare della cuspidi con solo feed forward

## 5.6 Verifica correttezza calcolo spline

In questa sezione si riportano i risultati ottenuti applicando il controllo Nonlinear Design come validazione della correttezza della traiettoria costruita a partire dai

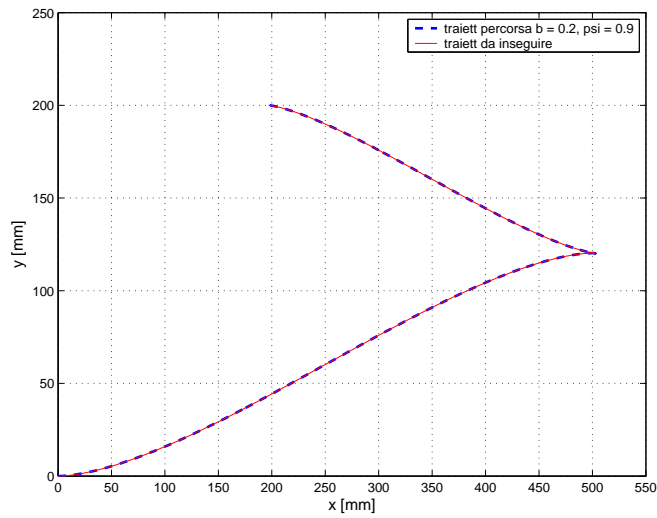


Figura 5.15: Inseguimento traiettoria con cuspidi (controllo Nonlinear Design)

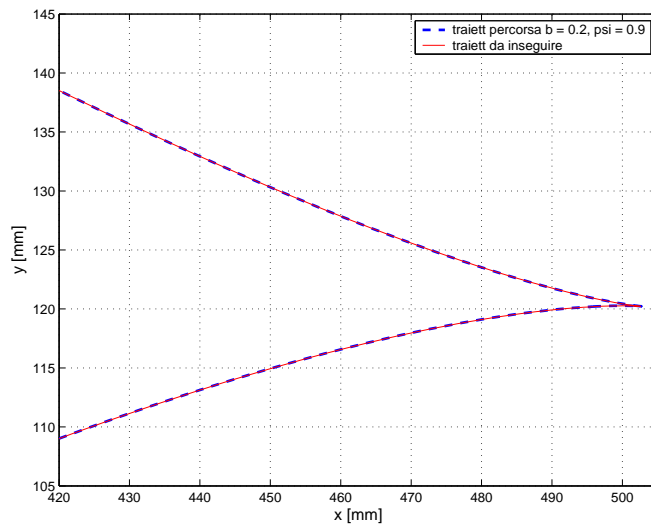


Figura 5.16: Particolare della cuspidi con controllo Nonlinear Design

soli punti di passaggio a velocità costante. Come si vede in figura 5.17 l'inseguimento risulta sovrapposto alla traiettoria desiderata, e l'errore è nell'ordine dei  $10^{-3}[mm]$ , come riportato in figura 5.18

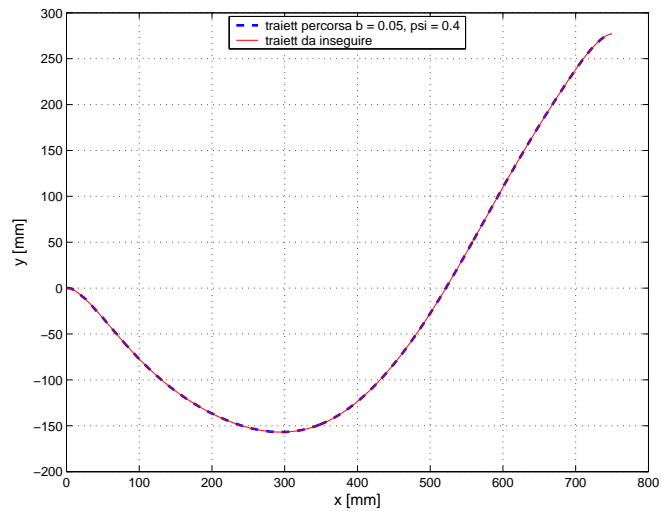


Figura 5.17: Inseguimento della traiettoria generata con spline

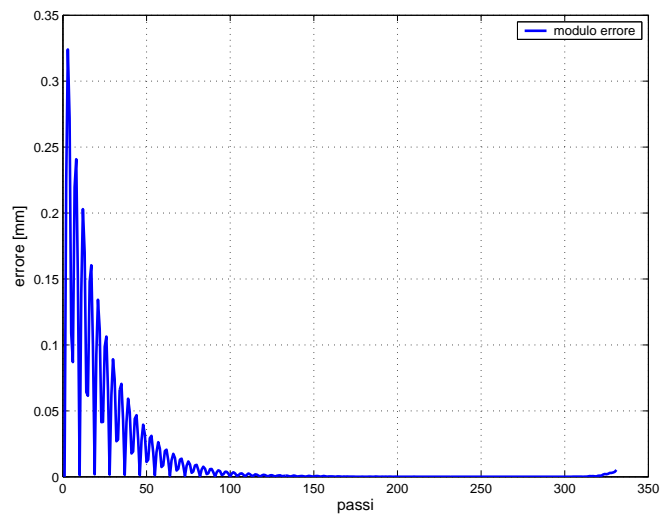


Figura 5.18: Errore d'inseguimento della spline



# Capitolo 6

## Risultati sperimentali

In quest'ultima fase si riportano i risultati ottenuti in fase sperimentale nello stesso ordine del capitolo precedente, non prima di aver illustrato le principali problematiche riscontrate in sede d'opera.

### 6.1 Problematiche sperimentali

I problemi principali riscontrati sono relativi a:

- applicazione controllo real time
- limiti di velocità dell'attuatore e controllo PID integrato
- determinazione posizione baricentro a partire dai valori degli encoder

La prima problematica è sorta in quanto il tempo di calcolo e di comunicazione è risultato tutt'altro che trascurabile, come già anticipato, e più precisamente dell'ordine dei  $40[ms]$ , che applicando dei margini di sicurezza ha portato alla scelta del tempo di campionamento di  $60[ms]$ .

In secondo luogo, le velocità applicabili alle ruote, per limitare lo slip longitudinale, sperimentalmente sono risultate non superiori ai  $100[mm/s]$ . Considerazione, questa, che è già stata implicitamente utilizzata nella scelta delle traiettorie già descritte. Inoltre, essendo presente un controllore PID a bordo del robot, le velocità imposte sono soggette ad un tempo di assestamento legato ai parametri di controllo interni.

In ultimo, è necessario valutare la posizione del baricentro a partire dalla conoscenza della posizione degli encoder. Per far ciò si è ottenuta la lunghezza percorsa da ogni ruota (riscalando opportunamente i contatori) ed in seguito derivata per ottenere, rispettivamente, la velocità della ruota destra  $v_r$  e della sinistra  $v_l$ . Da queste invertendo le relazioni dell'equazione 4.1 e integrando secondo:

$$\dot{x}(t) = v(t)\cos(\vartheta(t)) \quad (6.1)$$

$$\dot{y}(t) = v(t)\sin(\vartheta(t)) \quad (6.2)$$

si ricava la posizione spaziale del baricentro, da confrontare con quella voluta.

## 6.2 Feedforward

Il grafico di figura 6.1 riporta in analogia con quanto detto precedentemente la dinamica del sistema reale ottenuta con la sola imposizione dell'azione feedforward sulla traiettoria dell'8 piccolo. E' particolarmente interessante il risultato ottenuto in quanto dimostra chiaramente la necessità di un'azione di controllo in retroazione come era naturale attendersi.

Si evidenzia un ulteriore limite del sistema reale nella discordanza di comportamento della ruota destra da quella sinistra alle basse velocità.

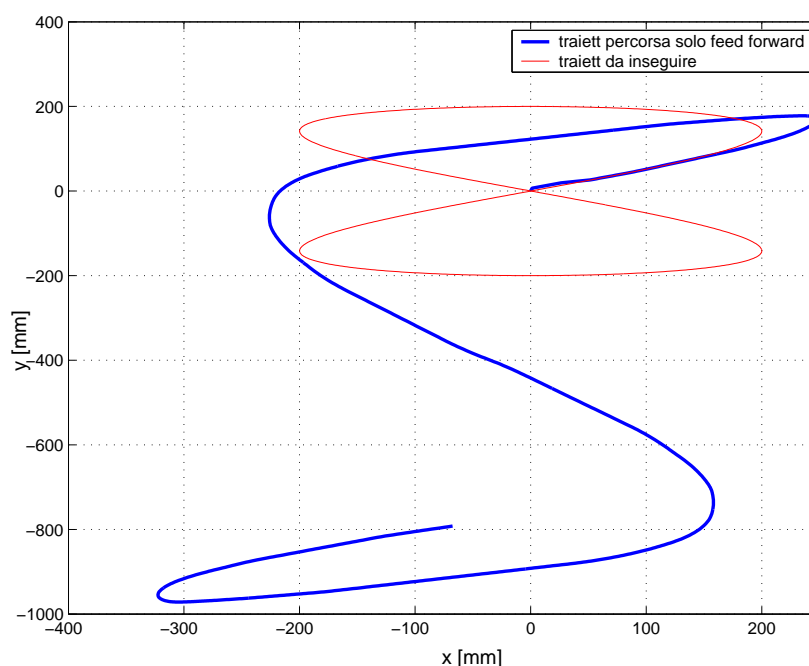


Figura 6.1: Inseguimento sperimentale con solo feed forward

## 6.3 Azioni di controllo

L'azione di controllo sperimentale è stata sviluppata, con le opportune modifiche, in modo analogo a quanto fatto in simulazione.

### 6.3.1 Linear design

I risultati ottenuti utilizzando le due traiettorie ad 8 di test sono riassunti nei due grafici degli errori di inseguimento di figura 6.2 ( $b = 0.01$  e  $\zeta = 0.4$ , valori ottenuti sfruttando i grafici di tuning ottenuti in simulazione) e 6.3 ( $b = 0.003$  e  $\zeta = 0.4$ ) dove si vede come, diversamente da quanto ottenuto in simulazione, con velocità maggiori si introducano errori maggiori (caso dell'8 grande). Come nel precedente capitolo, anche qui, si limiterà il confronto tra le due traiettorie a questa sezione, e si analizzerà la sola traiettoria di minor dimensione.



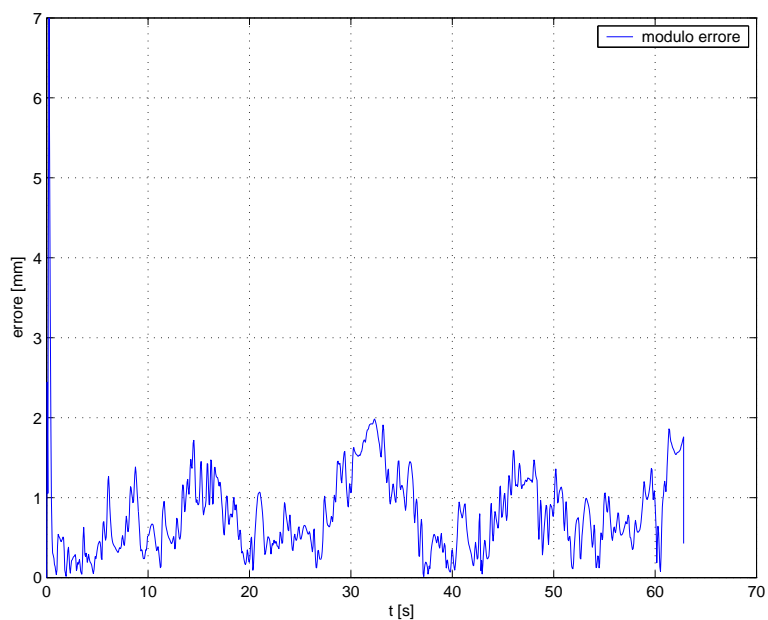


Figura 6.2: Errore su 8 piccolo con Linear Design

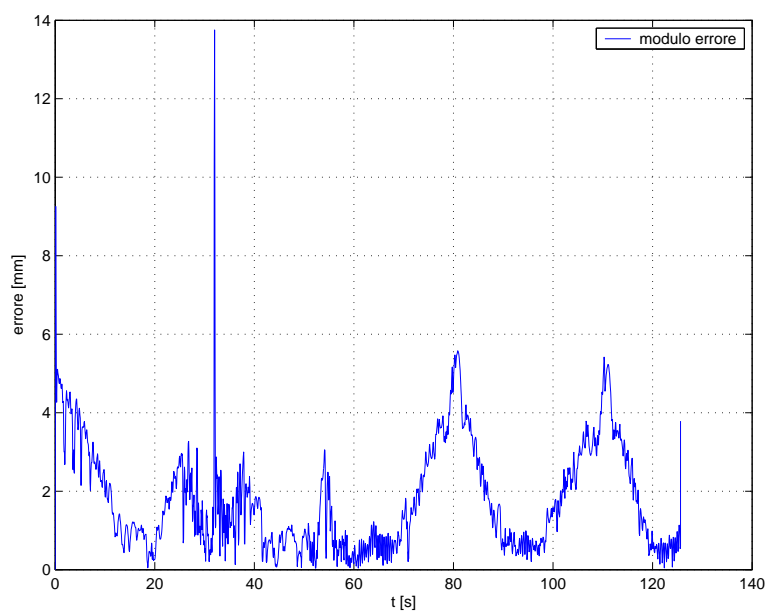


Figura 6.3: Errore su 8 grande con Linear Design

### 6.3.2 Nonlinear desing

L'errore risultante presentato in figura 6.4 ( $b = 0.005$  e  $\zeta = 0.3$ ) è analogo a quello ottenuto applicando il controllo lineare, confermando le similitudini tra i primi due controlli.

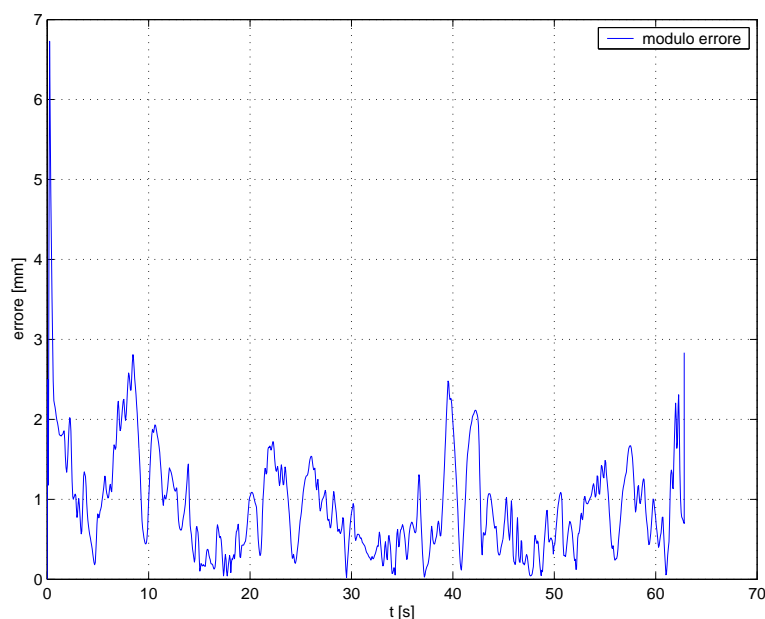


Figura 6.4: Errore su 8 piccolo con Nonlinear Design

### 6.3.3 Dinamic feedback linearization

Anche in questo caso si è preso spunto dal grafico per il tuning in simulazione per tarare i parametri del controllore, ottenendo come valori 'ottimi'  $k_p = 1.5$  e  $k_d = 0.3$ . L'errore risultante (visibile in figura 6.5) è sensibilmente maggiore di quello ottenuto nei casi precedenti.

Se ne conclude, dunque, un peggior comportamento sia in simulazione che in sperimentazione, non giustificando la maggiore complessità di progettazione del controllo.

## 6.4 Limiti del controllo

Nel caso si cerchi di ottenere prestazioni migliori, come in simulazione, si ottiene un comportamento oscillatorio nell'intorno della traiettoria desiderata. Come esempio si riporta in figura 6.6 il percorso compiuto utilizzando il nonlinear design con  $b = 0.01$  e  $\zeta = 0.3$ .

Nel caso di partenza con condizione angolare iniziale nulla (diversa da quella desiderata  $\vartheta = \text{atan}(0.5)$ ), il controllo permette di recuperare agevolmente questa mancanza d'informazione come si vede in figura 6.7.

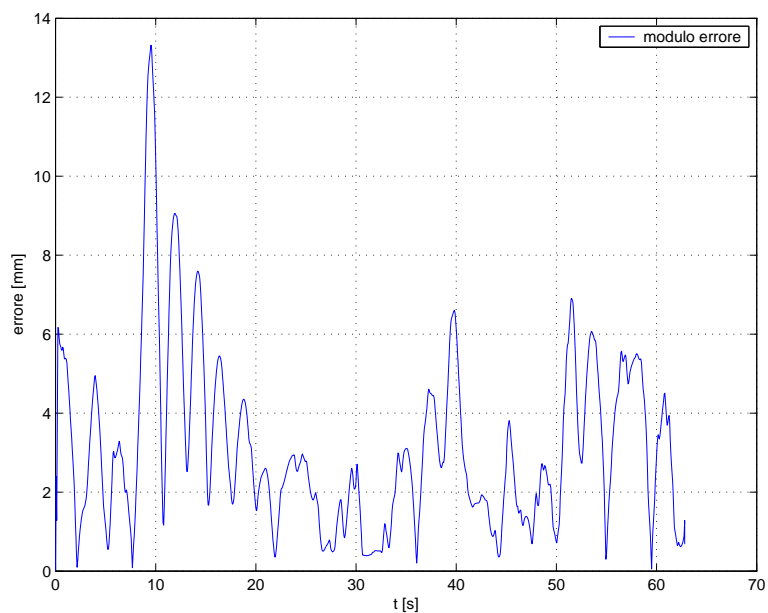


Figura 6.5: Errore su 8 piccolo con Dinamic Feedback Linearization

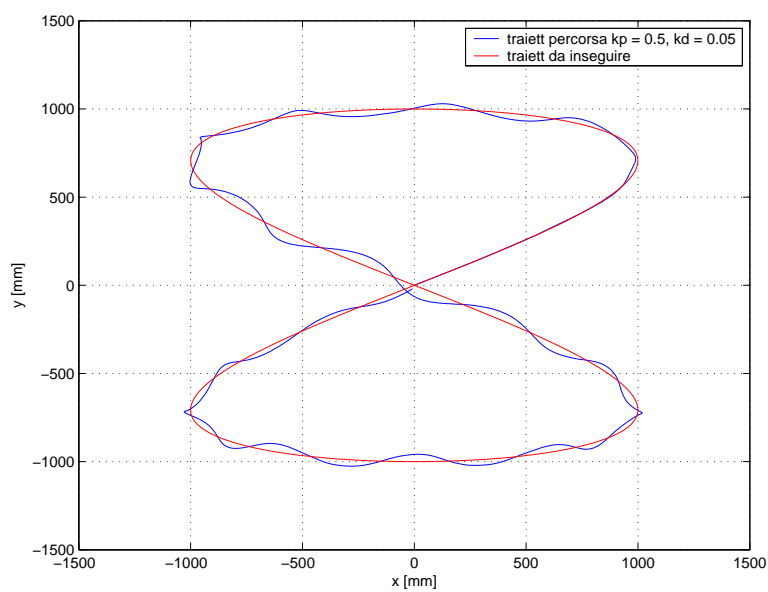


Figura 6.6: Inseguimento traiettoria 8 grande con controllo eccessivo

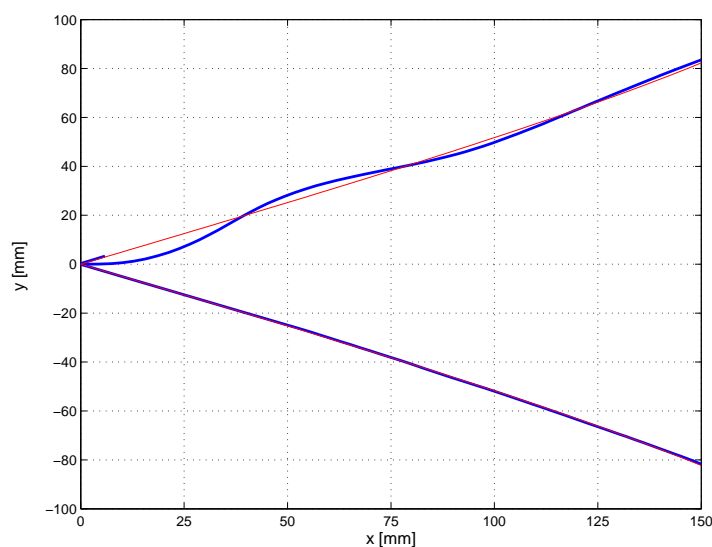


Figura 6.7: Partenza in spermentazione con orientazione errata

## 6.5 Cuspidi

Nell'ultimo caso presentato, riguardante l'inseguimento di una traiettoria punto-punto generata con ingressi sinusoidali che presenta una cuspidi, è stato applicato il controllo non lineare ed il risultato è esposto in figura 6.8. Come si può notare, anche in fase sperimentale si è riusciti ad inseguire correttamente la cuspidi, garantendo quindi, almeno con i primi due controlli, una vasta gamma di tipologie di traiettorie. Chiaramente, anche in questo caso non si è riusciti ad applicare il controllo basato sul modello dinamico.

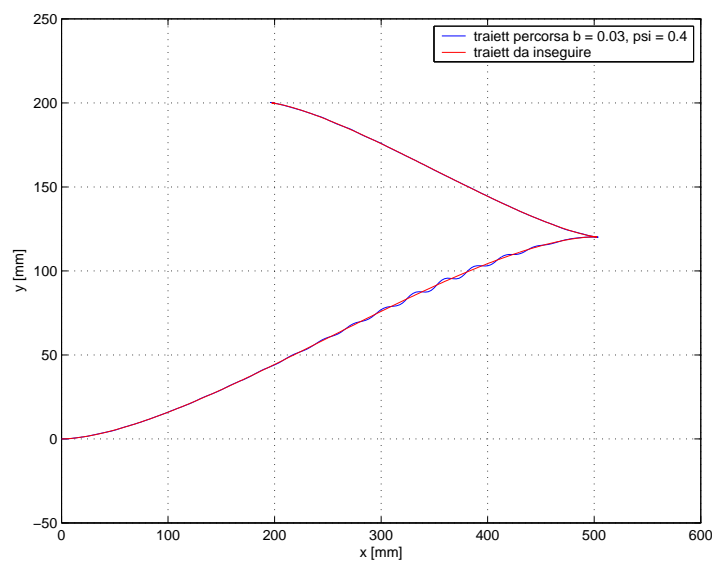


Figura 6.8: Inseguimento sperimentale di traiettoria punto-punto generata con ingressi sinusoidali

# Capitolo 7

## Evoluzioni future

Il progetto compiuto presenta notevoli punti di partenza per lavori futuri sia per progetti per il corso di 'Progettazione di sistemi di controllo' che possibili lavori di tesi. Alcune delle possibili evoluzioni sono descritte di seguito.

Acquisizione della posizione del robot da una sorgente esterna, quale può essere una telecamera o una rete di sensori, per poter evitare i problemi che possono insorgere dalla lettura degli encoder, che a causa dello slip sulle ruote può non fornire al sistema di controllo l'esatta posizione del sistema. Questa fonte esterna di acquisizione può essere agevolmente integrata in un anello di controllo più esterno a bassa frequenza, permettendo un'acquisizione più efficiente della posizione.

La pianificazione di traiettorie ottime in termini di tempo di percorrenza o consumo energetico non sono banali anche presupponendo un veicolo ideale. Nel lavoro di tesi eseguito da Macchion è stato valutato un aspetto simile che comportava la determinazione di traiettorie ottime per l'esecuzione dell'ELK-test (Single Line Change) per veicoli dell'Audi. La soluzione del problema è stata eseguita per via sperimentale, lanciando il modello del veicolo su un fascio di molteplici spline (un esempio è posto in figura 7.1) al fine di ottenere quella con minore tempo di percorrenza o maggiore velocità minima lungo il tracciato (in base alla discriminazione eseguita). Questo può essere uno spunto per possibili applicazioni analoghe sul robot Khepera.

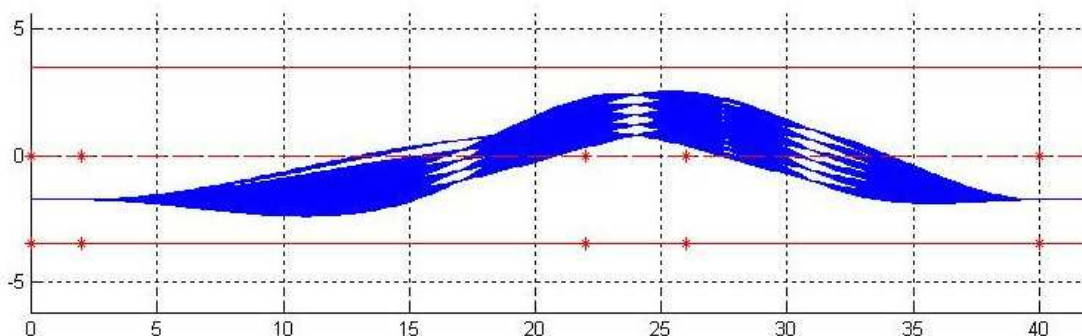


Figura 7.1: Spline generate per la simulazione



# Conclusioni

Il progetto è risultato particolarmente articolato sia dal punto di vista teorico che dal punto di vista applicativo. Innanzitutto la presenza di un sistema non oloonomo non ha permesso di utilizzare alcuna tecnica legata ai sistemi lineari e ha richiesto un'analisi approfondita del sistema per ottenerne un modello valido ai fini della simulazione e della progettazione del controllo. I risultati di maggior interesse sono risultati sicuramente

- la possibilità di generare una traiettoria con velocità imponibile interpolante una serie di punti nello spazio
- la generazione di traiettorie punto punto con ingressi polinomiali sul sistema a catena
- la possibilità di sfruttare gli encoder per effettuare controlli su un'ampia gamma di traiettorie con errori limitati senza slip

La prima caratteristica permette con alcuni accorgimenti di generare traiettorie in percorsi con ostacoli sfruttando ad esempio un accurato studio delle spline interpolanti. Lo sfruttamento del modello a catena con ingressi polinomiali ha messo in luce un'ottima capacità di generazione di traiettorie efficienti con basse velocità e avendo sviluppato la possibilità di imporre un tempo di percorrenza a piacere si è riusciti a dare un'enorme flessibilità a tale algoritmo.

L'errore di inseguimento che si è ottenuto con il solo utilizzo degli encoder è sicuramente incoraggiante ai fini di sviluppare un controllore basato su un'ulteriore acquisizione video esterna che sicuramente renderebbe più robusto il controllo estendendolo al caso di slittamento.

L'apparato sperimentale ha infatti presentato una serie di problemi legati alla comunicazione con il controllore (PC) e all'attuatore (motori) e al suo controllo embedded.





# Bibliografia

- [1] G. Oriolo, A. De Luca e M. Vendittelli, *WMR Control Via Dynamic Feedback Linearization: Desing, Implementation, and Experimental Validatio*
- [2] G. Oriolo, *Control of Nonholonomic Systems*
- [3] C. Guarino Lo Bianco, *Modellistica dei sistemi non olonomi*
- [4] Murray and Sastry, *Nonholonomic motion planning*
- [5] Luca Schenato, *Appunti delle lezioni del corso di progettazione di sistemi di controllo*, Anno Accademico 2005-06